

DISEÑO E IMPLEMENTACIÓN DE PIZARRAS DIGITALES DE BAJO COSTE

Blanco Gómez, Valentín
Sampedro Ruiz, Raúl

GRADO EN INGENIERÍA EN INFORMÁTICA. FACULTAD DE INFORMÁTICA
UNIVERSIDAD COMPLUTENSE DE MADRID



Trabajo Fin de Grado en Ingeniería Informática

Madrid, 19 de junio de 2014

Director/es y/o colaborador/es:

Gómez, José Ignacio
Piñuel, Luis

Autorización de difusión

Blanco Gómez, Valentín
Sampedro Ruiz, Raúl

Madrid, a 19 de junio de 2014

Los abajo firmantes, matriculados en el Grado de Ingeniería Informática de la Facultad de Informática, autorizan a la Universidad Complutense de Madrid (UCM) a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a su autor el presente Trabajo Fin de Grado: “DISEÑO E IMPLEMENTACIÓN DE PIZARRAS DIGITALES DE BAJO COSTE”, realizado durante el curso académico 2013-2014 bajo la dirección de Gómez, José Ignacio y la codirección de Piñuel, Luis en el Departamento de Arquitectura de Computadores y Automática, y a la Biblioteca de la UCM a depositarlo en el Archivo Institucional E-Prints Complutense con el objeto de incrementar la difusión, uso e impacto del trabajo en Internet y garantizar su preservación y acceso a largo plazo.



Esta obra está bajo una
Licencia Creative Commons
Atribución-NoComercial-CompartirIgual 4.0 Internacional.

*“There are only two kinds of languages: the ones people complain about
and the ones nobody uses.”*

B. Stroustrup

Agradecimientos

Gracias a Germán González Díaz del departamento de Física aplicada III por su infinita paciencia y su incalculable ayuda y asesoramiento en materia de electrónica analógica.

Índice general

Índice	I
Índice de figuras	IV
Índice de cuadros	VI
Resumen	VIII
Abstract	IX
1. Introducción	2
1.1. Concepto de pizarra digital	2
1.2. Nuevos dispositivos 3D	4
1.3. Propuesta de exploración	5
2. Situación actual	6
2.1. Pizarras digitales	6
2.2. Dispositivos 2D/3D	9
3. Evaluación de soluciones existentes	12
3.1. <i>Wiiimote</i>	12
3.2. <i>Kinect</i>	15
4. Desarrollo de solución propia	18
4.1. Aplicación pizarra digital	18
4.1.1. Reconocimiento gestual	19
4.1.2. Estructura de la aplicación	20

4.1.3.	Consideraciones al uso del <i>Leap Motion</i>	21
4.1.4.	Manual de uso de la aplicación	23
4.2.	Diseño del lápiz digital	27
4.2.1.	Estrategias de posicionamiento	27
4.2.2.	Medición de distancia de un emisor a un sensor	28
4.2.3.	Seguimiento de la posición del emisor	29
4.2.4.	Nuestro lápiz digital	30
4.2.5.	Nuestros sensores de distancia	31
4.3.	Uso del <i>Wii mote</i> en la aplicación de pizarra digital	34
5.	Conclusiones	36
5.1.	Acerca del coste de fabricación	36
5.2.	Sobre la viabilidad del posicionamiento de nuestra solución	39
5.3.	Sobre el uso en las aulas	40
	Bibliografía	41
	A. Circuitos	42
	B. Diagramas UML de la aplicación pizarra	44
B.1.	Diagramas UML de clases	44
B.2.	Diagramas UML de interacción	47
B.3.	Diagrama UML de dependencias entre paquetes	48
	C. Reparto de trabajo	50
C.1.	Blanco Gómez, Valentín	51
C.2.	Sampedro Ruiz, Raúl	54
	D. Introduction	58
D.1.	Digital whiteboard concept	58

D.2. New 3D devices	60
D.3. Proposed exploration	61
E. Conclusions	62
E.1. About the manufacturing cost	62
E.2. About viability of our positioning solution	65
E.3. About using digital whiteboards in classrooms	66

Índice de figuras

1.1. Pizarra digital en la <i>CeBIT</i> de 2007 (CC BY 2.0 http://www.flickr.com/photos/svonog/)	3
1.2. Controlador <i>Leap Motion</i> (CC BY-SA 3.0 http://commons.wikimedia.org/w/index.php?title=User:MorePix)	5
2.1. Vista frontal y trasera del <i>Wii mote</i> (CC BY-SA 3.0 http://ja.wikipedia.org/wiki/User:Oh-moo)	9
3.1. Soporte e instalación para usar un <i>Wii mote</i> como cámara infrarroja.	13
3.2. Emisor de infrarrojos en una placa de prototipado.	13
3.3. Instalación del <i>Wii mote</i> en el techo (CC BY-NC-SA 3.0 http://wyxs.net/web/wiiscan/)	14
4.1. Gesto desplazamiento a una mano (<i>swipe</i>) para el <i>Leap Motion</i>	21
4.2. Gesto círculo para el <i>Leap Motion</i>	22
4.3. Gesto toque de pantalla (<i>screen tap</i>) para el <i>Leap Motion</i>	22
4.4. Gesto toque de tecla (<i>key tap</i>) para el <i>Leap Motion</i>	23
4.5. Pantalla principal de la aplicación.	24
4.6. Pantalla del modo pizarra.	25
4.7. Pantalla del modo de selección de lienzos.	26
4.8. Representación de los frentes de onda esperados en el <i>receptor</i>	30
4.9. Diagrama de bloques de los osciladores del lápiz digital.	31
4.10. Instalación propia del <i>Wii mote</i> como sensor del lápiz digital y del <i>Leap Motion</i> como sensor 3D.	35

5.1. Prototipo de sensor implementado.	37
5.2. Prototipo de lápiz implementado.	38
A.1. Circuito de lápiz emisor.	42
A.2. Circuito receptor para <i>STM32</i>	43
B.1. Diagrama UML de clases del paquete <i>leapmotion</i>	44
B.2. Diagrama UML de clases del paquete <i>blackboard</i>	45
B.3. Diagrama UML de clases del paquete <i>protoboard</i>	46
B.4. Diagrama UML de interacción del paquete <i>blackboard</i>	47
B.5. Diagrama UML de interacción del paquete <i>protoboard</i>	48
B.6. Diagrama UML de dependencias entre paquetes de la aplicación.	48
D.1. Interactive whiteboard at <i>CeBIT</i> 2007 (CC BY 2.0 http://www.flickr.com/photos/svonog/)	59
D.2. <i>Leap Motion</i> device (CC BY-SA 3.0 http://commons.wikimedia.org/w/index.php?title=User:MorePix)	60
E.1. Implemented prototyped sensor.	63
E.2. Implemented prototyped pencil.	64

Índice de cuadros

4.1. Relación de frecuencia de muestreo con máxima distancia medible.	30
---	----

Resumen

En la actualidad, las pizarras interactivas se están convirtiendo en un elemento habitual en las aulas de cualquier centro educativo. Abren la puerta a nuevas estrategias de enseñanza que permiten a los estudiantes interactuar de manera diferente con contenido educativo y participar activamente, fuertemente motivados por el uso de nuevas tecnologías. Sin embargo, el precio de las soluciones comerciales suele sobrepasar los mil euros, haciendo imposible su uso en áreas con pocos recursos económicos.

Nuestro trabajo tiene dos objetivos principales: primero, desarrollar una pizarra interactiva funcional reutilizando tecnologías existentes con el fin de demostrar que se puede lograr a un precio mucho más bajo; en segundo lugar, extender su funcionalidad tradicional añadiendo sensores 3D. En nuestra exploración, hemos descubierto que el *Wii mote* es un punto de partida excelente para implementar una pizarra básica y barata. También hemos desarrollado nuestro propio sistema de posicionamiento usando sensores ultrasónicos y de infrarrojos, construyendo un prototipo cuyo coste está por debajo de los 50€. Ambas soluciones permiten hacer de un simple proyector una pizarra interactiva funcional y completa usando un lápiz de infrarrojos como sustituto de la *anticuada* tiza. Por otra parte, hemos integrado el sensor 3D *Leap Motion* en un nuestro software de pizarra digital que reconoce una serie de gestos y los traduce a eventos de ratón: *scrolling*, *zoom*...

Uniando ambas soluciones (posicionamiento y sensores 3D) podemos crear una pizarra interactiva realmente potente por aproximadamente 150€ (sin incluir el coste del proyector).

Palabras clave

Pizarra interactiva, sensor 3D, dispositivo de interfaz humana, posicionamiento 2D, tra-

tamiento de señal, reconocimiento gestual, microcontrolador.

Abstract

Nowadays, interactive blackboards are becoming a must in any educational center. They open the door to a new teaching strategies that allow students to interact in a different way with the presented content, as they get highly motivated through the use of new technology. However, the prices of current commercial systems are in the range of thousands of euros, preventing widespread usage in low-income areas.

The goal of this work is twofold: first, developing a functional interactive blackboard reusing existing technology in order to show that it may be achieved at a much lower price and, secondly, extend its traditional functionality by using the recently launched 3D sensors. In our exploration, we have found that the *Wii mote* is an excellent, cheap starting point for a basic whiteboard. We have also developed our own basic positioning system using ultrasonics and infrared sensors, building a prototype whose cost would be well below 50€. Both solutions allows to make a simple projector become a fully functional interactive whiteboard by using a infrared pen as a substitute of the *old-fashioned* piece of chalk. Moreover, we have integrated the *Leap Motion* 3D sensor in a home-made whiteboard software emulator which recognizes several hand gestures and translates them into effective mouse actions: scrolling, zooming...

By bringing together both solutions (positioning and 3D sensing) we may provide a very powerful interactive blackboard solution from 150€ (not including the projector itself).

Keywords

Interactive whiteboard, 3D sensor device, human interface device, 2D positioning, signal processing, gesture recognition, microcontroller.

Capítulo 1

Introducción

En este capítulo vamos a definir los conceptos de pizarra digital, explorando también sus usos, funcionalidad e inconvenientes. Además introduciremos los nuevos dispositivos de posicionamiento y detección de objetos y manos, y describiremos una propuesta de exploración en base a esta información.

1.1. Concepto de pizarra digital

Una pizarra digital interactiva consiste, a grandes rasgos, en un ordenador y una superficie sobre la que se mostrará la salida de video (pantalla táctil, proyector tradicional...). La característica distintiva de una pizarra digital es la existencia de algún mecanismo de detección que permita controlar el ordenador (generalmente, el puntero del ratón) interactuando con la propia imagen proyectada. Para llevar a cabo dicha interacción, normalmente se utiliza algún tipo de bolígrafo específico o incluso el dedo. El objetivo fundamental será reemplazar una pizarra tradicional, ampliando notablemente su funcionalidad.

El ámbito de uso de una pizarra digital es muy diverso: espacios de trabajo en grupo, reuniones, conferencias... Pero, a día de hoy, sin duda el uso más habitual es en el ámbito docente, cualquier que sea el nivel del alumnado.

Los múltiples usos que se les pueden dar incluyen[7]:



Figura 1.1: Pizarra digital en la CeBIT de 2007.

- Ejecutar software en el ordenador para ser controlado mediante la pizarra digital (por ejemplo, un navegador web).
- Escribir y editar notas en la pizarra que quedarán guardadas en el PC.
- Mostrar notas escritas con una tableta gráfica.
- Controlar el ordenador desde la pizarra usando acciones “arrastrar y soltar” que pueden producir, por ejemplo, anotaciones en diferentes aplicaciones o presentaciones. Esto abre un abanico enorme de posibles usos docentes, desde educación infantil hasta la universidad.

Estas pizarras digitales *clásicas* presentan una serie de inconvenientes: tienen una funcionalidad “limitada” pues su uso normalmente se reduce a mapear la acción del usuario con movimientos o clicks del ratón en el ordenador. Otro inconveniente es que los sensores utilizados tienen un rango 2D, de manera que no es posible capturar movimientos interesantes como podría ser un gesto de manos ya que estos sensores se limitan a una superficie plana. Por último, otro gran inconveniente es el precio: en función de la tecnología en cuestión, existe una gran variedad de precios, generalmente altos. Los sistemas más asequibles suelen

proporcionar únicamente los mecanismos de detección. Por ejemplo, encontramos un sistema que combina sensores infrarrojos y de ultrasonidos a un precio de 559€¹, pero no incluye superficie de proyección ni proyector. Si buscamos sistemas que incluyan la pizarra, incluyendo pantallas táctiles muy grandes, el precio aumenta considerablemente. Como ejemplo, una pizarra completa basada en tecnología infrarroja duplica el precio anterior (1245\$²); una pantalla táctil completa de 65" es aún más cara (4173€³). Además en muchos de estos casos hay que tener en cuenta que suelen requerir una instalación específica, que impide llevarlas fácilmente de un sitio a otro; en el caso de colegios o universidades esto implica la necesidad de múltiples unidades.

1.2. Nuevos dispositivos 3D

La aparición de nuevos tipos de sensores que no se limitan a un campo 2D sino que permiten detectar un mapa de profundidad utilizando sensores 3D, abre nuevas vías de desarrollo para estas pizarras digitales. Su funcionalidad se amplía notablemente, permitiendo detectar gestos, movimientos naturales del usuario como “pasar una página” o “hacer *zoom* a una imagen”. Como ejemplos podemos poner el sensor *Kinect* de *Microsoft*, que proporciona información de profundidad, distinción de personas y gestos, y está pensado para ser colocado en una distancia media de forma que detecte todo el cuerpo. Recientemente se comercializó el sistema *Leap Motion* pensado, en un principio, para una interacción más cercana al usuario (cerca del teclado del ordenador) y permite detectar principalmente gestos en profundidad 3D de las manos. Se hará un análisis más detallado de estos dispositivos en la sección 2.2 y en el capítulo 3.

¹<http://www.mimioonline.com/p/1/Mimio%20Teach/MimioTeach/>

²<http://www.goelectronic.com/QOM0+QWB200-PS.html>

³<http://pizarras-digitales.net/pantalla-interactiva-65-4-points>



Figura 1.2: *Controlador Leap Motion.*

1.3. Propuesta de exploración

El objetivo del proyecto es el diseño de un sistema que permita convertir un proyector en una pizarra digital. Así, a partir de los datos anteriores, elaboramos una propuesta de exploración para evaluar distintas soluciones con distinta funcionalidad y analizando su coste. Para ello nos centramos en tres aspectos fundamentales:

- **Dispositivos:** vamos a analizar el uso de diferentes dispositivos y tecnologías con las que conseguir interaccionar con el sistema (posicionamiento, acciones tipo *click*, arrastrar....).
- **Coste:** tendremos que valorar las distintas soluciones procurando siempre minimizar el coste total del prototipo, para conseguir un sistema lo más barato posible.
- **Funcionalidad:** valoraremos soluciones y dispositivos que aporten nueva funcionalidad a la idea de pizarra digital, como por ejemplo el uso de sensores de mapa de profundidad 3D.

Asimismo, habrá que diseñar el software necesario (a ser posible, multiplataforma) para que el elemento de posicionamiento usado actúe de forma efectiva como ratón en el ordenador. También consideraremos el desarrollo de aplicaciones que exploten la funcionalidad del sistema.

Capítulo 2

Situación actual

En este capítulo analizamos el estado actual de las pizarras digitales en el mercado, basadas en sus diferentes tecnologías, con ejemplos comerciales en cada caso. También analizamos el estado de los distintos dispositivos de *tracking* 2D/3D actuales, incluyendo también ejemplos comerciales y precios.

2.1. Pizarras digitales

Existen diversas formas de interacción con el usuario que implican distintas tecnologías y distintos costes[7]:

- **Basada en tecnología infrarroja:**

Un sistema de infrarrojos se coloca en la superficie de la pizarra de manera que los movimientos de los dedos u otras herramientas (bolígrafos, rotuladores...) son capturados gracias a la interferencia que generan en el campo infrarrojo. De ese modo, el software puede determinar la posición del objeto mediante triangulación.

Esta tecnología proporciona una funcionalidad 2D que permite utilizar la posición de un objeto para *mapear* clicks o clicks continuos del ratón dentro del ordenador.

Como ejemplo comercial el sistema *QOMO QWB200-PS 78"* (una pizarra completa) que tiene un precio de 1245\$¹, o el sistema *Tidvision Ticboard 80"* (una pizarra

¹<http://www.goelectronic.com/QOMO+QWB200-PS.html>

completa) que tiene un precio de 886€².

- **Basada en tecnología electromagnética:**

Es la versión de las tabletas gráficas a gran escala: consiste en una serie de cables colocados detrás de la superficie de la pizarra y un bolígrafo pasivo que altera la señal producida por ésta, determinándose así la posición del bolígrafo. La herramienta puede tener también un botón en la punta, para que, al ser apretado contra la superficie, simule un click del ratón. Nos proporciona una funcionalidad 2D que puede simular tanto movimiento del ratón (cuando el bolígrafo está *cerca* de la superficie), como clicks del ratón en algunos casos.

Como ejemplo comercial el sistema *QOMO QWB300EM 88"* (una pizarra completa) que tiene un precio de 1995\$³, o el sistema *PROMETHEAN ACTIVBOARD 378 PRO 78"* (una pizarra completa) que tiene un precio de 1119€⁴.

- **Basada en tecnología resistiva:**

Toda la superficie de la pizarra digital se compone de un sistema táctil resistivo común: una membrana exterior que se deforma mediante presión hace contacto con un material conductor. El punto de presión se puede localizar entonces electrónicamente. El objeto utilizado puede ser cualquier herramienta, y, como en el caso anterior, nos proporciona una funcionalidad 2D que registra clicks del ratón.

Como ejemplo comercial el sistema *TeamBoard TMWM7450EM 85"* (una pizarra completa) que tiene un precio de 1795\$⁵.

- **Basada en tecnología capacitiva:**

Como el sistema electromagnético, se basa en unos cables colocados detrás de la superficie. Sin embargo, es la interacción con los dedos (y no con un bolígrafo especial)

²<http://pizarras-digitales.net/tidvision-ticboard-80-pizarra-interactiva-multitactil-4p>

³http://www.touchboards.com/qomo/qomo_qwb300em.html

⁴<http://pizarras-digitales.net/pizarra-digital-interactiva-promethean-activboard-378-pro>

⁵http://www.touchboards.com/teamboard/teamboard_tmwm7450em.html

la que permite proporcionar coordenadas. Como es de esperar, al igual que en la tecnología electromagnética, nos puede proporcionar una funcionalidad 2D que simular movimiento y clicks del ratón.

Como ejemplo comercial el sistema *Promethean 578 Pro ActivBoard 78"* (una pizarra completa) que tiene un precio de 2059£⁶.

- **Basada en tecnología infrarroja + ultrasónica:**

Con esta tecnología, se usa un apuntador especial que emite infrarrojos y ultrasonidos. Conociendo las diferentes propiedades de las ondas empleadas, se calcula la diferencia entre la velocidad de la luz y la del sonido, lo que permite localizar el puntero. Esto proporciona un sistema de detección 2D para el posicionamiento del puntero.

Como ejemplo comercial encontramos el sistema *MimioTeach* (sólo el detector y bolígrafo) que tiene un precio de 559€⁷.

- **Basada en tecnología de proyector interactivo:**

Un proyector interactivo (además de ser un proyecto normal) contiene una cámara CMOS que detecta la posición de un bolígrafo infrarrojo activo cuando éste entra en el umbral de la imagen proyectada. Puede tener problemas como la intromisión del propio cuerpo de la persona entre la cámara del proyector y el bolígrafo, que se puede arreglar con los actuales proyectores de distancia ultracorta, que emiten normalmente de forma vertical y se colocan en la parte de arriba de la pizarra, de forma que el cuerpo no puede interferir en la señal.

Nos proporciona una funcionalidad de detección 2D que, en principio, sólo puede simular clicks o un click continuo de ratón.

Como ejemplo comercial el sistema *Epson EB-485Wi* que tiene un precio de 2208€⁸.

⁶<http://www.misco.co.uk/product/188760/Promethean-ActivBoard-578-Pro>

⁷<http://www.mimioonline.com/p/1/Mimio%20Teach/MimioTeach/>

⁸<http://pizarras-digitales.net/epson-proyector-eb-485wi-interactivo>

2.2. Dispositivos 2D/3D

La aparición de nuevos tipos de sensores que permiten detectar un mapa de profundidad utilizando sensores 3D y de dispositivos de detección 2D de bajo costo abren nuevas vías de investigación para el desarrollo de pizarras electrónicas de bajo coste:

■ *Wiiote:*

Es el mando principal de la consola *Wii* de *Nintendo*. Su característica más destacable es la posibilidad de utilizar su cámara infrarroja incorporada para detectar y posicionar un punto de emisión de luz infrarroja[10]. Ofrece, por tanto, detección y posicionamiento 2D. Además es un dispositivo barato (19€⁹).



Figura 2.1: *Vista frontal y trasera del Wiimote.*

■ *Kinect:*

Es un controlador diseñado por *Microsoft* para la videoconsola *XBOX 360*, pero puede ser usado para otros fines. Tiene cámaras de profundidad e infrarrojas que, combinado con los controladores disponibles, proporcionan, por ejemplo, información de profundidad, distinción de personas y gestos. Esta funcionalidad permitiría adaptarlo para emular una pizarra digital interactiva[9]. Cuesta unos 140€¹⁰.

⁹<http://www.amazon.es/Nintendo-4902370517842-Wiimote/dp/B002C1AUPA/>

¹⁰<http://www.amazon.es/Microsoft-Sensor-Kinect-Xbox-360/dp/B004RFC940/>

- ***Leap Motion:***

Es un sensor que soporta *tracking* de manos, dedos y gestos en profundidad 3D. Está pensado para una interacción cerca del usuario (en el teclado del ordenador). Ofrece detección y posicionamiento en 3D. Tiene un precio de 99€¹¹.

- **Dispositivos de ultrasonidos / infrarrojos:**

Los dispositivos de ultrasonidos son ampliamente utilizados para el cálculo de distancias por rebote como se usan por ejemplo en los sistemas de asistencia de aparcamiento. El uso de estas ondas sonoras es amplio porque no son percibidas por el oído humano[8] y por tanto no molestan a quien esté cerca del transmisor.

La emisión de la onda sonora se puede combinar con una onda electromagnética (como la luz), de manera que en un determinado punto alejado del foco de emisión se puede detectar el desfase en la recepción de las dos ondas, ya que la luz viaja a mayor velocidad que el sonido.

Al igual que los ultrasonidos no son percibidos por el oído humano, la luz infrarroja no es visible al ojo humano[6], lo que la convierte en una solución atractiva de cara a evitar molestias.

¹¹<http://www.amazon.es/Leap-Motion-LM-C01-DE-Controlador-movimiento/dp/B00H45P892/>

Capítulo 3

Evaluación de soluciones existentes

En este capítulo se presentan con más detalle algunos de los dispositivos expuestos en la sección 2.2, evaluándolos como posibles candidatos para diseñar una pizarra electrónica. Para cada uno de ellos se ha realizado un montaje que permite constatar sus pros y contras a la hora de utilizarlos como pizarra digital.

3.1. *Wiiimote*

Hemos partido del trabajo desarrollado por D. Schouten[10], adaptando el código proporcionado a nuestro contexto de uso. Hemos empleado un ordenador, un proyector común, una superficie plana rígida donde proyectar la imagen del ordenador, un soporte para el *Wiiimote*, y un bolígrafo emisor de infrarrojos. Para el soporte del mando adaptamos un trípode común para que pudiera sujetar un *Wiiimote*, como se aprecia en la fig. 3.1. Como bolígrafo utilizamos un led emisor de infrarrojos prototipado en una placa (fig. 3.2).

Hemos analizado diferentes aspectos de esta solución:

- **Precisión**

Alta. Permite escribir perfectamente de manera legible siempre y cuando se respete la visibilidad entre la persona y el *Wiiimote*. La cámara infrarroja del mando tiene un aspecto de 4:3, y dado que tiene un ángulo de visión horizontal de 40° y vertical



Figura 3.1: *Soporte e instalación para usar un Wiimote como cámara infrarroja.*

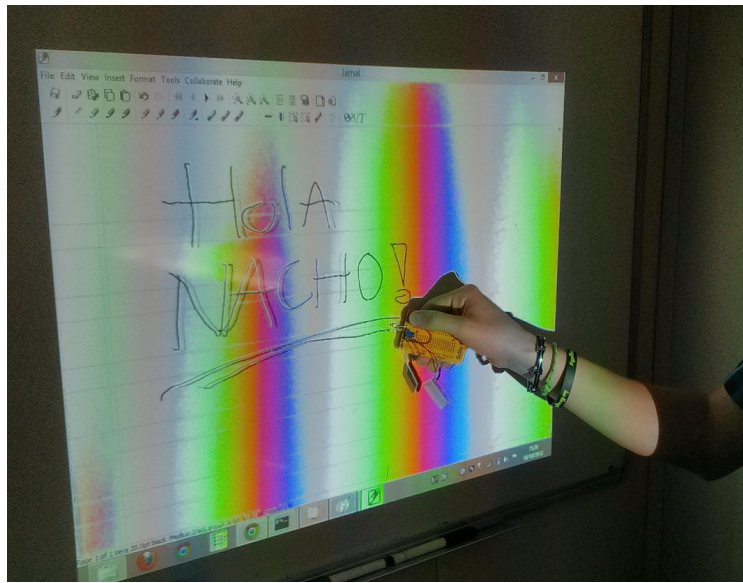


Figura 3.2: *Emisor de infrarrojos en una placa de prototipado.*

de 30° [10] puede colocarse sin problemas en el techo de la habitación de manera que apunte de forma oblicua a la superficie de proyección y sea muy complicado interferir con el propio cuerpo entre la señal que emite el bolígrafo infrarrojo y la cámara receptora (fig. 3.3).



Figura 3.3: *Instalación del Wiimote en el techo.*

■ Manejabilidad

Alta, aunque limitada por la tecnología del bolígrafo pues es poco intuitivo tener que apretar un botón para escribir, si bien existen modelos de bolígrafo infrarrojo que se activan al apretarse contra una superficie¹. Además está el problema de la visión del *Wiimote*, que hace que no sea muy intuitivo su uso si no está “bien colocado”. Para asegurar una buena visión del *Wiimote* podemos, o bien colocar éste colgado en el techo con un ángulo contrapicado agudo, o colocar éste a la altura de la pizarra, formando un ángulo de hasta 40° con el plano de la pizarra en el lado de la mano que se use para escribir.

■ Fiabilidad

La aplicación en sí no es muy robusta e impone alguna restricción, como la necesidad de apretar los dos botones del *Wiimote* durante la inicialización. Esto implica que el *Wiimote* debe ser fácilmente accesible al comienzo del uso de la pizarra o bien montar un sistema de pinza que presione ambos botones todo el tiempo, lo que reduce el

¹<http://www.ebay.com/itm/Black-Infrared-IR-Pen-Wiimote-Remote-Interactive-Whiteboard-Activated-Slim/111297898895>

tiempo de vida del dispositivo ya que el *Wiimote* es un sistema con batería autónoma que debe reemplazarse. No es muy realista pensar en ponerlo en el techo junto al proyector sin dotarle antes de un sistema de alimentación fiable y continuo (que no requiera cambiar o mover el mando).

Sin embargo, es posible solventar esa limitación tal y como se describe en el capítulo 3 (*“Building the USB power module & Wiimote powering”*) de [10]: es factible construir un módulo para alimentar el *Wiimote* mediante USB. Con esto sería posible no tener que sustituir las baterías cuando se acaben (con la correspondiente recalibración de la posición del mando), complicando el cableado de la habitación en la que se instale el sistema.

■ Tecnología de interacción

La aplicación simula un puntero; cuando el *Wiimote* detecta un punto de luz infrarroja dispara un evento de click de ratón que no se libera hasta perder el punto de luz, por lo que nos permite hacer click y doble click pero no click con el botón derecho del ratón o *scrolling* de manera nativa.

3.2. *Kinect*

En esta categoría hemos podido evaluar una solución comercial de *Ubi Interactive*². Ésta coloca el *Kinect* a la altura del proyector, mirando hacia la pantalla. *Ubi* usa el mapa de profundidad del *Kinect* para detectar las pulsaciones en una superficie plana, estableciendo un umbral de profundidad de disparo con respecto a una calibración inicial, que consiste en una imagen patrón a pantalla completa que es proyectada y capturada por el *Kinect* para establecer los límites de la pantalla, y un mapa de profundidad para poder fijar el umbral de activación. Cuando un objeto se acerca más de ese determinado umbral al fin del mapa de profundidad de la calibración, se detecta que el usuario está tocando la pantalla, y con

²<http://www.ubi-interactive.com/>

la información de posición obtenida con la calibración el gesto se traduce en una pulsación de ratón.

- **Precisión**

Baja. El uso de la mano entera como puntero hace completamente inviable la escritura con esta solución. Resulta difícil realizar acciones que requieran un mínimo de precisión y su uso queda limitado a interfaces de botones grandes o muy gestuales.

- **Manejabilidad**

Media. La visibilidad que tiene el *Kinect* de la pantalla es muy buena sin necesidad de colocarlo en posiciones elevadas o ángulos cerrados. Por desgracia el sistema de detección de pulsaciones no se comporta muy bien, dando lugar a falsas pulsaciones.

- **Fiabilidad**

La aplicación es muy estable. El *Kinect* posee alimentación externa lo que nos asegura que no se van a causar interrupciones por falta de alimentación.

- **Tecnología de interacción**

La aplicación simula un dispositivo de entrada/salida táctil de tal manera que permite la detección de hasta tres puntos, permitiendo gestos de pinza (con dos manos), *scrolling*, click (tocando), click izquierdo (toque largo). Como la aplicación es de código cerrado, todo lo que podamos aventurar sobre su implementación son conjeturas.

Capítulo 4

Desarrollo de solución propia

En este capítulo seguimos el desarrollo de una solución propia de pizarra electrónica. En ella se implementa un reconocimiento gestual basada en tecnología de sensores 3D, y un posicionamiento del lápiz digital basado en infrarrojos y ultrasonidos.

4.1. Aplicación pizarra digital

Para diseñar una aplicación que emule el funcionamiento de una pizarra digital hemos utilizado el lenguaje de programación *Java*, para generar una aplicación multiplataforma. También utilizamos dos librerías externas: *Processing* y la *Leap Motion SDK Java library*. Para proveer de reconocimiento gestual al software, hemos utilizado dispositivos sensor 3D *Leap Motion* (sección 2.2).

Hemos considerado el uso de *Processing* para el desarrollo de la aplicación por ser una librería que facilita la producción de bajo coste de prototipos multimedia e interactivos de diseño digital.

Para el reconocimiento gestual, hemos considerado el uso de un dispositivo *Leap Motion* por tener una gran relación precio / potencial de uso.

Todo el código generado en esta parte está disponible en un repositorio online en *GitHub*¹.

¹<http://github.com/valenbg1/protoboard>

4.1.1. Reconocimiento gestual

El *API* para programar el *Leap Motion* provee de multitud de facilidades para la detección de gestos, posicionamiento de manos, dedos y herramientas. Toda la parte que interactúa con el *Leap Motion* está englobada en la clase *leapmotion.LeapMotionListener* (diagrama UML de las clases del paquete *leapmotion* disponible en la figura B.1), que extiende *Listener* que es una clase que provee el *API* para poder recoger los eventos que manda el dispositivo. Para el diseño del código y la programación del *Leap Motion* también se ha utilizado como referencia el libro “*Leap Motion Development Essentials*”, escrito por M. Spiegelmock[11].

LeapMotionListener mantiene un conjunto de “observadores” que implementan la interfaz *leapmotion.LeapMotionObserver*, y serán llamados cuando *LeapMotionListener* detecte eventos interesantes (patrón *Observer*). En nuestra aplicación, la parte que implementa la pizarra digital (clase *blackboard.Blackboard*) y la parte que implementa el mapeo de expresiones gestuales a eventos de teclado y ratón (clase *input.Input*) son observadores de los eventos que manda *LeapMotionListener*.

La parte interesante de la clase *LeapMotionListener* es la función sobreescrita *onFrame(Controller controller)*, que es llamada cada vez que el controlador tiene disponible nueva información de *tracking* (recoge información periódicamente). Básicamente recoge el *frame* que le manda el dispositivo y si no es un movimiento traslación (que no está implementando directamente en el *API*), pasa a comprobar si es alguno de los cuatro gestos detectados de serie por el *API*: círculos, desplazamiento a una mano (*swipe*), toque de pantalla (*screen tap*) y toque de tecla (*key tap*). Si es alguno de estos cuatro, manda el evento correspondiente a los observadores.

En el caso de un movimiento de traslación (que hemos implementando como dos manos juntas moviéndose), hemos implementado el reconocimiento de un nuevo gesto. Comprobamos que en el *frame* actual y en el pasado se detectan dos manos cercanas y que son las mismas,

medimos el vector que resulta de trasladar el estado anterior al actual, y este vector se manda a los observadores mediante el evento *onTranslation(float d_x, float d_y)* (en mm).

4.1.2. Estructura de la aplicación

La aplicación en sí se compone de tres paquetes principales: *blackboard*, *input*, *leapmotion* y *swing*.

La estructura del paquete *leapmotion* ya ha sido explicada en la sección 4.1.1, y se encarga de la interfaz con el dispositivo *Leap Motion* y del reconocimiento gestual.

El paquete *blackboard*, cuya descripción del diagrama de clases se encuentra en la figura B.2, y del de interacción entre clases en la figura B.4, se encarga de implementar la función de pizarra digital. La clase principal, *Blackboard*, extiende *PApplet*, una clase que provee la librería de *Processing*. Básicamente mantiene una lista de *PGraphics*, que son las distintas pantallas de la aplicación que el usuario puede ir pasando, gestiona la interacción del usuario con ellas y implementa las diversas opciones que da el interfaz (de cara al propio usuario).

El paquete *input*, cuya descripción del diagrama de clases se encuentra en la figura B.3, se encarga de implementar el *mapeo* de los eventos que manda *LeapMotionListener* a eventos de teclado y ratón.

El paquete *swing* contiene las clases que definen la interfaz principal del programa, que está diseñada en *Swing*.

La parte principal de la aplicación y punto de llamada inicial, la clase *Main*, se encuentra descrita mediante un diagrama de clases UML en la figura B.3, y su diagrama de interacción en la figura B.5. Mantiene un objeto *Blackboard* y un objeto *Input* y implementa las inter-

acciones entre éstos, así como crea la interfaz principal y maneja los eventos que el usuario puede mandar desde ésta.

Por último, el diagrama de dependencias entre paquetes de la aplicación puede observarse en la figura B.6.

4.1.3. Consideraciones al uso del *Leap Motion*

El *Leap Motion* debe estar colocado en la parte de abajo de la pizarra, y un poco separado de ésta, como se puede observar en la figura 4.10. En la sección 1.2 se comenta que el dispositivo está pensado para una interacción cercana al usuario, el rango de detección que provee el *Leap Motion* es una pirámide invertida con la punta en el propio dispositivo. Por lo tanto, para un reconocimiento óptimo de los gestos, éstos deben realizarse encima del *Leap Motion* pero no demasiado alejados de éste.

Los gestos detectados por defecto en la API del *Leap Motion* son:

- **Desplazamiento a una mano (*swipe*):** en las cuatro direcciones.

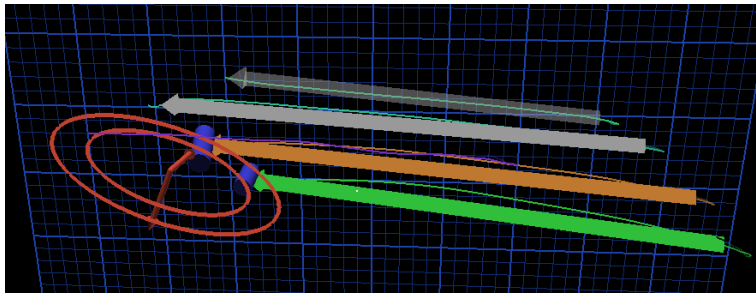


Figura 4.1: Gesto desplazamiento a una mano (*swipe*) para el *Leap Motion*.

- **Círculo:** realizado con un dedo, en sentido horario / antihorario.

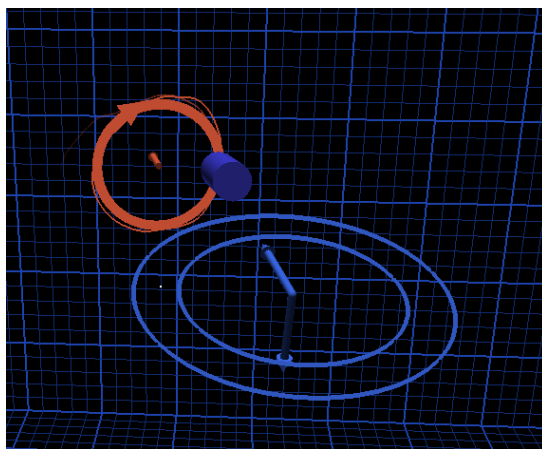


Figura 4.2: *Gesto círculo para el Leap Motion.*

- **Toque de pantalla (*screen tap*):** un toque realizado con el dedo sobre una superficie vertical imaginaria.

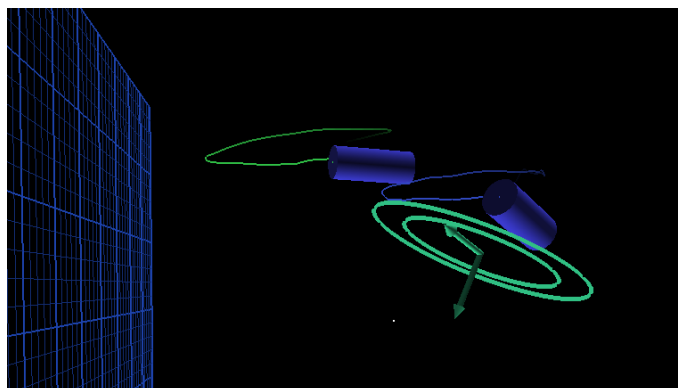


Figura 4.3: *Gesto toque de pantalla (*screen tap*) para el Leap Motion.*

- **Toque de tecla (*key tap*):** un toque realizado con el dedo sobre una superficie horizontal imaginaria.

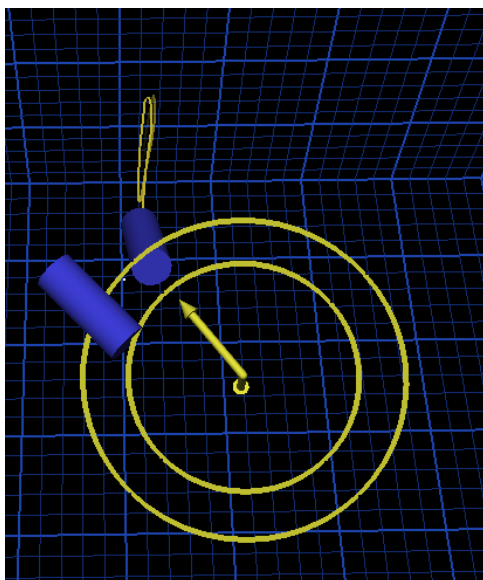


Figura 4.4: *Gesto toque de tecla (key tap) para el Leap Motion.*

4.1.4. Manual de uso de la aplicación

Al iniciar la aplicación nos muestra una interfaz principal que se compone de dos botones y una serie de menús, como se observa en la figura 4.5.

En las opciones de menú encontramos:

- ***File:***

En este menú podemos cargar una serie de imágenes *.png* guardadas previamente con la aplicación, guardar todas las imágenes que estén actualmente en el modo pizarra en una carpeta, o salir.

- ***Configuration:***

En este menú podemos acceder a los menús de opciones de los dos modos, pizarra y *mapeo*, donde podemos elegir entre otras cosas la sensibilidad del *Leap Motion* o la carpeta por defecto donde se guardarán las imágenes de la aplicación.

- ***Help:***

En este menú podemos acceder a una pantalla de información sobre los autores del programa así como información de contacto.

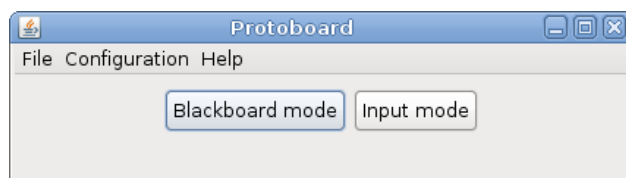


Figura 4.5: *Pantalla principal de la aplicación.*

Los dos botones principales nos llevan a los modos principales de la aplicación.

En el modo pizarra (*Blackboard mode*, como se observa en la figura 4.6) disponemos de tres selectores que controlan las funciones principales. El primero arriba a la izquierda permite seleccionar el tamaño del puntero para pintar, con las flechas izquierda / derecha. El selector de la esquina inferior izquierda permite seleccionar el color del puntero, y también seleccionar el modo borrar, con las flechas izquierda / derecha. El selector de la derecha permite alternar entre distintas pantallas en la aplicación, pudiendo tener varios lienzos distintos, con las flechas izquierda / derecha. Si pulsamos en la flecha hacia arriba, nos muestra una pantalla de selección donde podemos navegar por los distintos lienzos activos en ese momento, como se observa en la figura 4.7. Si pulsamos en la flecha hacia abajo, guarda el lienzo actual en una imagen *.png* en el directorio elegido en las opciones.

Por otra parte, los gestos que podemos realizar con el *Leap Motion* en el modo pizarra son los siguientes:

- **Desplazamiento a una mano (*swipe*):**

El desplazamiento a una mano a izquierda / derecha cambia el lienzo actual hacia delante / hacia atrás.

El desplazamiento a una mano abajo guarda el lienzo actual / quita el modo de selección de lienzos.

El desplazamiento a una mano arriba activa el modo de selección de lienzos.

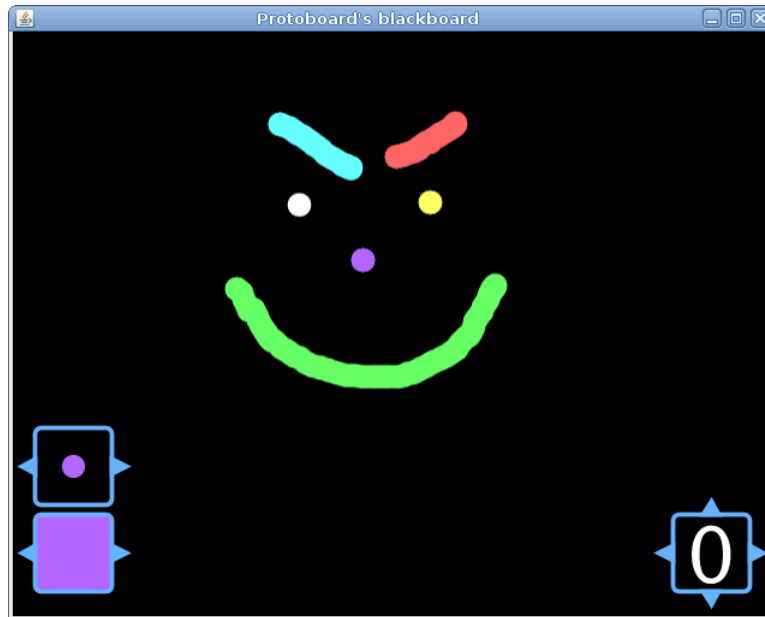


Figura 4.6: *Pantalla del modo pizarra.*

- **Círculo:**

Un círculo en el sentido antihorario / horario cambia el color del puntero hacia atrás / hacia delante.

- **Toque de pantalla (*screen tap*) / toque de tecla (*key tap*):**

Seleccionan el lienzo actual en el modo de selección de lienzos.

- **Traslación:**

Permite desplazarse en las cuatro direcciones dentro del lienzo actual, y en el modo de selección de lienzos (muestra un recuadro rojo que indica la porción de lienzo que se está mostrando).

En el modo que implementa el *mapeo* de expresiones gestuales a eventos de teclado (*Input mode*), podemos interactuar con otras aplicaciones mediante gestos detectados por el *Leap Motion* (las teclas que se *mapean* pueden configurarse en la pantalla principal):

- **Desplazamiento a una mano (*swipe*):**

El desplazamiento a una mano a izquierda / derecha presiona las flechas derecha /

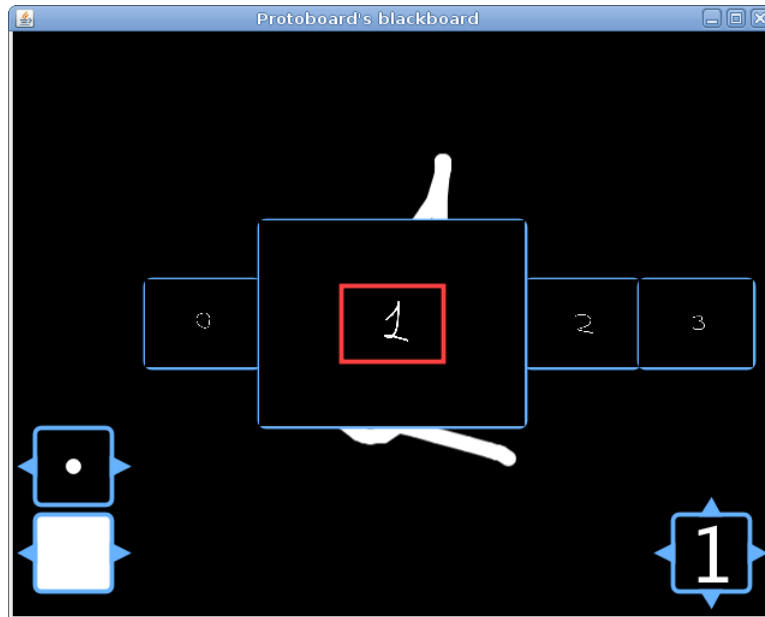


Figura 4.7: *Pantalla del modo de selección de lienzos.*

izquierda.

El desplazamiento a una mano arriba presiona F11.

Estos gestos se pueden usar, por ejemplo, para pasar diapositivas o activar el modo pantalla completa en una presentación.

- **Círculo:**

Un círculo en el sentido antihorario / horario presiona las flechas izquierda / derecha.

Estos gestos se pueden usar para pasar diapositivas rápidamente.

- **Toque de pantalla (*screen tap*) / toque de tecla (*key tap*):**

Presionan el botón izquierdo del ratón.

- **Traslación:**

Hacia arriba / abajo mueve la rueda del ratón abajo / arriba.

Este gesto se puede usar, por ejemplo, para desplazarse por un documento.

4.2. Diseño del lápiz digital

Con una aplicación que emula el funcionamiento de una pizarra ya funcionando, aún queda implementar una solución *hardware* para poder emular el ratón en forma de lápiz digital. Para esta tarea hemos decidido usar una aproximación de muy bajo coste con sensores de ultrasonido e infrarrojos, para evaluar si es factible la elaboración de una solución completa y si ésta daría una precisión adecuada para su uso.

4.2.1. Estrategias de posicionamiento

La medida de distancia de un objeto baliza a una estación se puede realizar de muchas maneras. Por ejemplo en los sistemas GPS[5], un satélite envía un mensaje al cliente con un *timestamp* y la posición del satélite en el momento de envío del mensaje. Como el mensaje tarda un tiempo considerable en llegar al cliente, éste puede estimar con gran precisión su distancia relativa al satélite. Si obtenemos estos mismos datos de otros dos satélites más, la intersección de las circunferencias que tienen como centro cada uno de los satélites y como radio la distancia relativa del cliente a ellos es con alta precisión la posición del cliente.

Por supuesto esta solución es cara de implementar y solo tiene sentido en un entorno de distancias largas, pero la base de la triangulación de la señal es similar en la mayoría de soluciones de medida de distancia absoluta. Nuestra solución es parecida aunque se simplifica mucho al conocer ciertos datos sobre el entorno. Para nosotros los *satélites* (en adelante *sensores* o *receptores*) están fijos en una posición relativa a la pantalla que conocemos, y están separados entre ellos a una distancia también conocida, no necesitamos que el *cliente* (en adelante *emisor*) sepa su posición, son los *receptores* los que tienen que saberla.

El hecho de conocer la posición relativa de los *emisores* a la pantalla y la distancia entre éstos hace que sólo necesitemos dos receptores para poder estimar la posición del *emisor*. Como no necesitamos que el *emisor* sepa nada sobre su posición, éste puede ser completamente

autónomo.

4.2.2. Medición de distancia de un emisor a un sensor

En la sección 2.2 ya vimos una primera aproximación sobre cómo combinar sensores y emisores de infrarrojos y ultrasonidos para poder estimar la distancia de un objeto a una estación. En esta sección vamos a desarrollar la base teórica de la solución.

La velocidad de la luz es 6 órdenes de magnitud mayor que la del sonido, lo que significa que si un emisor lanza una señal sonora y luminosa a la vez, la luminosa llegará considerablemente antes al destino que la sonora. Ambas señales han recorrido la misma distancia, por lo que podemos afirmar que:

$$c * t = v_s \cdot (\delta + t)$$

Donde c es la velocidad de la luz, v_s la del sonido, t el tiempo que tarda en llegar la señal luminosa (variable) y δ el retardo de la señal sonora.

Con esto podemos afirmar que:

$$t = \frac{v_s \cdot \delta}{c - v_s}$$

Y por tanto la distancia del emisor al receptor ha de ser:

$$d = \frac{v_s \cdot c}{c - v_s} \cdot \delta$$

Donde todos los datos son conocidos.

La constante $\frac{v_s \cdot c}{c - v_s}$ tiene un valor de 343000.39 mm/s tomando como v_s la velocidad del

sonido en condiciones estándar. Este valor es perfectamente aproximable a la propia velocidad del sonido con precisión milimétrica (más que suficiente para nuestra aplicación).

$$d \simeq \delta \cdot v_s$$

Nosotros por simplicidad trabajaremos con esta aproximación ya que no supone una pérdida apreciable de precisión.

4.2.3. Seguimiento de la posición del emisor

Para realizar un seguimiento del emisor sólo tenemos que conseguir que éste emita una señal periódicamente, pero la frecuencia con la que se emite la señal es muy importante en nuestro caso. Una frecuencia muy baja disminuiría la resolución del seguimiento haciendo que trayectorias curvas puedan parecer rectas. Sin embargo una frecuencia muy alta disminuye la distancia máxima de medición. La razón del primer problema es trivial. El segundo problema puede requerir algo más de explicación.

En la figura 4.8 podemos ver una representación ideal de los frentes de onda esperados en los sensores de infrarrojos y ultrasonidos del receptor respectivamente. δ representa el desfase del ultrasonido frente al infrarrojo y es la variable que, como se ha visto anteriormente (sección 4.2.2), nos permite calcular la distancia del emisor al receptor. Si el emisor y el receptor estuvieran a distancia cero, δ se anularía y los dos frentes de onda quedarían totalmente en fase.

T es el periodo con el que decidimos enviar las dos señales en el emisor. En éste, ambas señales se emiten en fase con el mismo periodo. El problema viene porque δ ha de estar acotado al menos por T , ya que si δ es mayor que T , no podemos determinar si el frente de onda detectado fue producido en el emisor junto con el primer frente de onda de infrarrojos

o junto con el segundo (y el anterior frente de ultrasonidos se ha perdido).

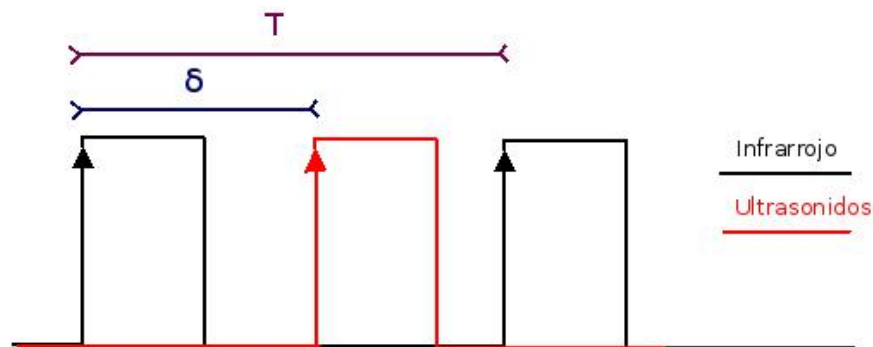


Figura 4.8: Representación de los frentes de onda esperados en el receptor.

Si δ ha de ser menor que T , entonces es evidente que a mayor frecuencia de muestreo menor $\max(\delta)$, y por tanto menor es la distancia máxima de medición.

En la tabla 4.1 podemos ver una serie de valores para la relación máxima distancia medible con frecuencia de muestreo.

f en Hz	$\max(distancia)$ en m
10	34.30
30	11.43
60	5.72
120	2.86

Cuadro 4.1: Relación de frecuencia de muestreo con máxima distancia medible.

Una frecuencia en el emisor de 30Hz o de 60Hz establece una relación perfecta entre resolución de muestreo y máxima distancia medible.

4.2.4. Nuestro lápiz digital

Con los resultados teóricos de los apartados anteriores, ya estamos en disposición de crear un prototipo de lápiz digital que pueda ser usado. Para la generación de los trenes

de pulsos usamos como integrados dos *LM555*, que son unos dispositivos altamente estables para la generación precisa de retrasos y oscilaciones[1].

Ambos *LM555* son configurados en modo *astable* (para ser usados como osciladores). Uno a 40KHz (frecuencia de oscilación del emisor de ultrasonidos), con una carga de ciclo de aproximadamente el 50 %, y otro a 30Hz (frecuencia que según la tabla 4.1 nos da una relación distancia máxima/muestreo muy aceptable) con una carga de ciclo de aproximadamente el 20 % (oscilador maestro).

El oscilador de ultrasonidos es habilitado por el maestro de tal manera que, cada $\frac{1}{30}s$ el lápiz emite un pulso de ultrasonidos e infrarrojos de aproximadamente $\frac{1}{150}s$. En la figura 4.5 podemos ver una simplificación en forma de diagrama de bloques de la configuración de los dos osciladores.

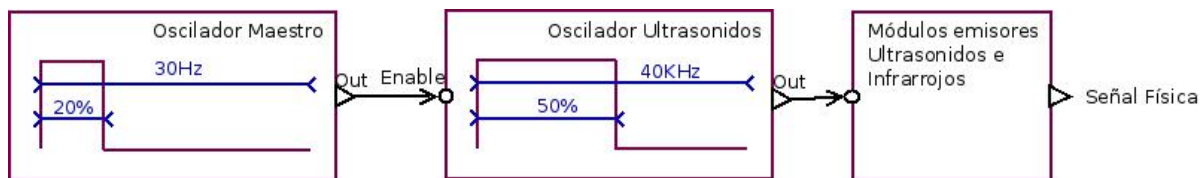


Figura 4.9: Diagrama de bloques de los osciladores del lápiz digital.

4.2.5. Nuestros sensores de distancia

Como se puede ver en la figura 4.4, tenemos que ser capaces de medir el desfase que sufre la llegada del frente de onda de la emisión de ultrasonidos con respecto a la de infrarrojos para poder determinar la distancia del lápiz a nuestro sensor. Para poder medir este desfase con precisión milimétrica es necesario un microcontrolador con temporizadores con una resolución de al menos 350KHz aproximadamente (ya que el sonido tarda unos $3\mu s$ en recorrer un milímetro). Actualmente no es difícil encontrar microcontroladores a muy bajo

precio del orden de los 10MHz de frecuencias de reloj.

En nuestro caso para los sensores hemos decidido usar un *STM32F100RB*[\[2\]](#) por varias razones :

- Es un microcontrolador, aunque relativamente sencillo, muy completo en funcionalidades.
- Su frecuencia máxima de operación es de 24MHz, más que de sobra para nuestro uso.
- Es fácilmente miniaturizable a un integrado de 5x5mm
- Es un microcontrolador de bajo coste.

Para el proceso de desarrollo hemos usado una placa con este microcontrolador (la *stm32vlDiscovery*) que está especialmente adaptada para éste. De todas las características que nos ofrece el *STM32F100RB*[\[2\]](#), en nuestro prototipo usamos:

- *GPIOs (General-purpose inputs/outputs ports)*: Puertos de entrada de señales para la captura de los flancos de ultrasonidos e infrarrojos.
- *EXTI (External Interrupt)*: Dos líneas de interrupción para la detección de los flancos, más otra para el temporizador de eliminación de rebotes.
- *Temporizadores*: Para medir el desfase entre flancos y otras medidas de control como la eliminación de rebotes o la sincronía en la transmisión.
- *USART (Universal synchronous/asynchronous receiver transmitter)*: Una conexión por puerto serie para comunicar la distancia medida al host donde se emula el ratón.
- *NVIC (Nested vectored interrupt controller)*: El controlador de interrupciones vectorizadas anidadas para ajustar las prioridades de cada interrupción.

A continuación vamos a pasar a describir los pasos que ha de llevar a cabo el micro-controlador para la obtención de un dato de medida en un contexto ideal (sin ruido y sin flancos perdidos, desordenados o duplicados):

- (1) Inicialización de los vectores de interrupción para la captura de flancos. Habilita solo la interrupción de infrarrojos.
- (2) Detección de flanco de infrarrojos:
 - a) Inicializa los temporizadores de medición y de transmisión.
 - b) Habilitar la interrupción de ultrasonidos y deshabilitar la de infrarrojos (evita ruido).
- (3) Detección de flanco de ultrasonidos:
 - a) Calcula distancia medida y prepararla para su transmisión.
 - b) Deshabilitar la interrupción de ultrasonidos (evita ruido).
- (4) Vencimiento del temporizador de transmisión:
 - a) Transmite el dato de medida obtenido por puerto serie.
 - b) Rehabilitar la interrupción de infrarrojos.
- (5) Esperar de nuevo a (2).

Hemos de tener en cuenta que los eventos de flancos o vencimiento de temporizadores no tienen por qué seguir la secuencia antes descrita. Si bien el vencimiento del temporizador de transmisión está forzado a ocurrir después de un flanco de infrarrojos, podría ser que no tuviera un flanco de ultrasonidos entre medias (porque éste se haya perdido). De la misma manera cualquiera de los flancos podría ser duplicado por efectos de rebote o ruido. Por esta razón, forzamos la serialización de las detecciones de los flancos deshabilitando y habilitando convenientemente las interrupciones.

Como se describió en la sección 4.2.3, la distancia máxima medible es indirectamente proporcional a la frecuencia de emisión del lápiz. Para asegurar esta restricción usamos el temporizador de transmisión. Cuando se detecta un flanco de infrarrojos este temporizador se inicia. El tiempo de vencimiento de este temporizador está programado según la frecuencia de emisión del lápiz, de tal manera que éste ha de vencer siempre antes del siguiente flanco de infrarrojos (dejando todo preparado para su detección). Si en el momento que vence este temporizador no disponemos del dato de medida, se asume que se ha perdido el flanco de ultrasonidos y el *host* es notificado por el puerto serie con un valor de medida inválido.

Debido a los exigentes requisitos temporales que nos impone el problema, hemos prescindido de la sobrecarga que supondría un sistema operativo y el desarrollo sobre la placa de prototipado *STM32F100RB*[2] ha sido completamente *bare metal*. El acceso a los dispositivos y gestión de interrupciones se ha realizado sin capas de abstracción de *hardware* ni librerías del fabricante, complicando significativamente la programación y depuración del sistema.

4.3. Uso del *Wiimote* en la aplicación de pizarra digital

Como alternativa a un desarrollo propio del lápiz digital, podemos combinar la aplicación pizarra mencionada en la sección 4.1 con la solución usando un *Wiimote* evaluada en la sección 3.1. De esta manera, podemos usar el *Wiimote* para posicionar el ratón, y el *Leap Motion* para reconocer gestos. Un posible montaje sería el apreciado en la figura 4.10.

Como vimos en la sección 2.2, el *Wiimote* usa una cámara para detectar un punto de luz infrarroja, mientras que el *Leap Motion* emite tres puntos de luz infrarroja para poder detectar objetos dependiendo del patrón de reflejo. En principio, podríamos tener un problema dado que ambos dispositivos operan usando la misma tecnología (infrarrojos). Esta problemática

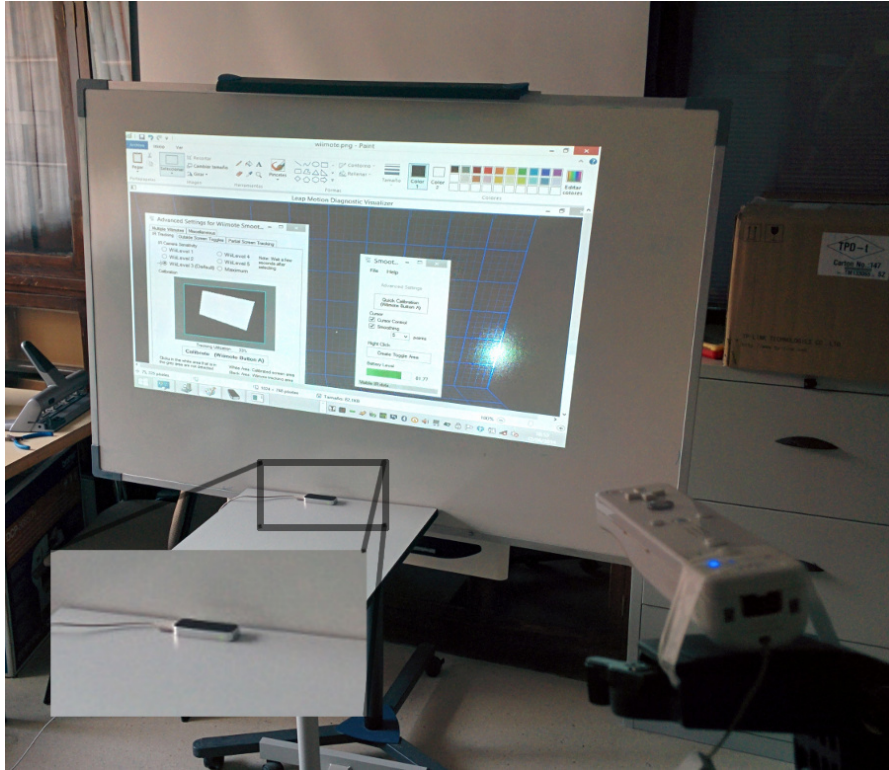


Figura 4.10: *Instalación propia del Wiimote como sensor del lápiz digital y del Leap Motion como sensor 3D.*

se resuelve colocando ambos dispositivos en ángulos de operación suficientemente separados, de forma que la luz emitida por el *Leap Motion* no interfiera en la detección del lápiz por el *Wiimote*. Esta restricción nos limita el ángulo de contrapicado al que se puede someter al *Wiimote*. Si decidimos colocar el *Wiimote* en el techo cerca de la pizarra con un ángulo contrapicado agudo para evitar el problema, mencionado en 3.1, de ocultar la visión de la cámara infrarroja con el propio cuerpo, podríamos observar problemas ocasionados por el ruido generado por el *Leap Motion*.

Capítulo 5

Conclusiones

En este proyecto hemos explorado diferentes alternativas para el desarrollo de pizarras digitales de bajo coste que, además, añadan funcionalidad a las existentes en el mercado. Hemos comprobado como, con la salida de los sensores 3D al mercado, se amplían notablemente las posibilidades de estos dispositivos a precios asequibles.

Utilizando uno de esos dispositivos (*Leap Motion*), hemos desarrollado un sistema capaz tanto de permitir la interacción con el escritorio como de emular una pizarra que permita guardar y compartir fácilmente todas las notas escritas.

En el ámbito de la pizarra digital *tradicional* (esto es, posicionamiento 2D del puntero del ratón), hemos estudiado alternativas de muy bajo coste incluyendo una de elaboración propia basada en tecnología de ultrasonidos e infrarrojos. En el resto de esta sección cualificaremos el coste de esta solución y discutiremos sobre su posible aplicación en entornos reales.

5.1. Acerca del coste de fabricación

Ya en el capítulo 2 veíamos la situación actual de las pizarras digitales y sus precios. Nuestro objetivo principal desde entonces ha sido intentar rebajar el precio lo máximo posible perdiendo un mínimo de calidad.

El coste en material al pormenor de uno de nuestros sensores como el que se ve en la figura 5.1 es de aproximadamente 23€. Su desglose aproximado es el siguiente:

- Placa STM32VDiscovery - 10€
- Sensor ultrasonidos 40Hz - 2€
- Sensor Infrarrojos - 2€
- Integrados (LM393 y TL081) - 3€
- Circuitería analógica - 1€
- Conversor DC-DC - 2€
- Cable USB a Serie TTL 3.3V - 3€

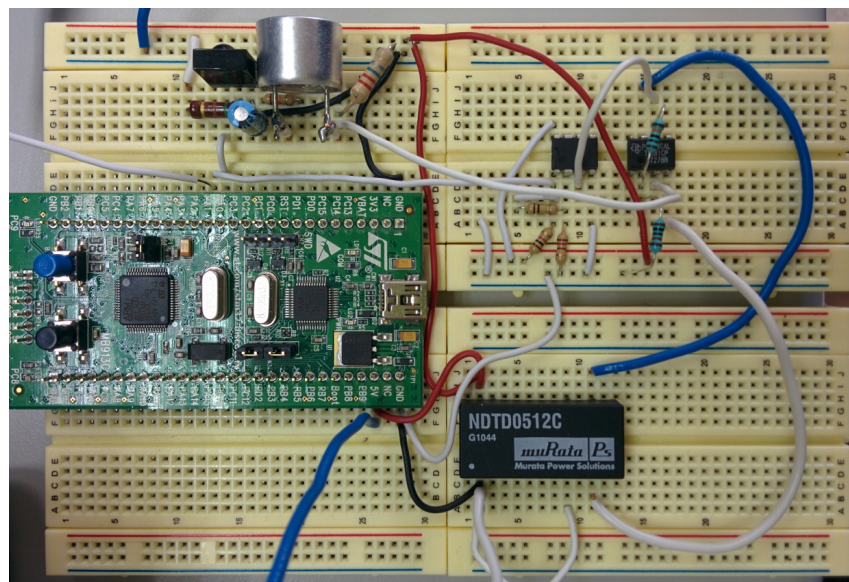


Figura 5.1: *Prototipo de sensor implementado.*

El elemento más caro de esta lista es la Placa STM32VDiscovery, que en realidad no necesitamos completamente. Bastaría con el procesador, que puede comprarse por separado por menos de 5€.

Nuestra solución para emular el ratón con estos sensores, como se veía en la sección 4.2.1, necesita de dos de estos sensores más un emisor (como el de la figura 5.2), cuyo coste no

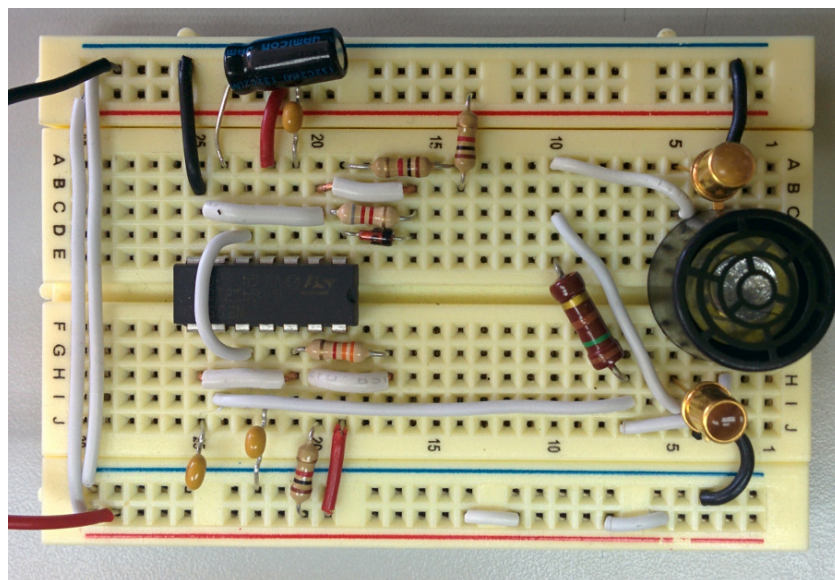


Figura 5.2: *Prototipo de lápiz implementado.*

supera los 4€ fabricado al por menor.

Si sumamos todos los costes podemos estimar un coste de unos 50€ en desarrollar el prototipo utilizado para posicionar el cursor. Si bien hay que tener en cuenta que bajo una comercialización al por mayor se reduciría muy notablemente el coste de fabricación, no podemos despreciar que el prototipo actual está pensado para demostrar la factibilidad del sistema, y una implementación comercial podría suponer un incremento de gastos con el objetivo de mejorar fallos actuales.

Si sumamos a esto el coste del *Leap Motion* (unos 90€), obtenemos una solución con reconocimiento gestual por apenas 140€, frente a los precios de las soluciones que vimos en el capítulo 2.

Hay que tener en cuenta que para estos cálculos no estamos suponiendo un coste añadido muy importante, que es el coste de un proyector y un ordenador para el uso del hardware y el software implementado. Este coste supera claramente al coste de los sensores, pero se

puede despreciar ya que (a parte de que muchas otras soluciones tienen este coste añadido) una instalación de proyección es muy común en cualquier aula. Aún así, de cara a minimizar el coste del ordenador, es muy factible portar la solución de ultrasonidos sin *Leap Motion* a una *Raspberry Pi*, por ejemplo.

5.2. Sobre la viabilidad del posicionamiento de nuestra solución

No muy lejos de una solución real, nuestro prototipo nos sirve para detectar problemas de implementación de la detección de posición por medio de ultrasonidos e infrarrojos. A continuación exponemos una lista de problemas explicados y sus posibles soluciones:

- **Direccionalidad del emisor de ultrasonidos:**

Desgraciadamente los emisores de ultrasonidos que hemos encontrado son altamente direccionales. Esto supone que si el emisor no apunta más o menos directamente hacia los sensores es relativamente fácil que se pierda la señal. Una posible solución es usar un emisor mas omnidireccional o incluso usar varios emisores apuntando en direcciones distintas.

- **Perdida de precisión ante ruido o atenuación:**

Ocasionalmente puede suceder que la primera oscilación del receptor de ultrasonidos se pierda o se atenúe retrasando el frente de onda de ultrasonidos $25\ \mu\text{s}$ (el tiempo de ciclo de los ultrasonidos). Es fácil ver que este pequeño retraso causa un error en la medida de aproximadamente 10 mm según las fórmulas vistas en la sección 4.2.2. Para solventar este pequeño error creemos que se puede o bien mejorar la etapa de amplificación en el receptor, o bien implementar un tratamiento a nivel de aplicación.

- **Bloqueo de señal por alumno malintencionado:**

Es relativamente fácil bloquear la señal del emisor de infrarrojos con otro emisor (como

por ejemplo un mando a distancia) correctamente dirigido hacia la barra de los sensores provocando la completa inutilización del sistema. Este problema es difícilmente tratable porque supone un canal con una relación señal / ruido muy baja.

5.3. Sobre el uso en las aulas

Durante el desarrollo del proyecto se han tenido reuniones con personal docente para obtener una serie de necesidades que ha de cumplir una pizarra digital. En esta sección relatamos algunas de ellas y como podrían cubrirse.

En el Colegio Público Asunción Rincón el personal docente dispone de un aula habilitada con pizarra digital. Esta tiene un cañón de proximidad y una superficie resistiva. El coste de su instalación ronda el orden de los 1500€. El personal docente usa estas pizarras para actividades interactivas con niños cómo por ejemplo juegos de relacionar figuras con nombres.

Para este tipo de actividad no es necesario una precisión muy elevada en el posicionado del puntero por lo que se podría implementar fácilmente con un sensor 3D como la *Kinect*.

En el Colegio Gredos San Diego están interesados en poder interactuar con la pizarra escribiendo en ella, de cara a poder guardar luego lo que se escribe. Este centro ha mostrado interés por nuestro planteamiento ya que les permitiría posicionar con suficiente precisión el cursor.

Bibliografía

- [1] *LM555 - Timer DataSheet*, 2000. <http://www.ti.com/lit/ds/symlink/lm555.pdf>.
- [2] *STM32F100RB - Microcontroller*, 2012. <http://www.st.com/st-web-ui/static/active/en/resource/technical/document/datasheet/CD00251732.pdf>.
- [3] *Leap Motion API reference*, 2014. http://developer.leapmotion.com/documentation/java/api/Leap_Classes.html.
- [4] *Processing API reference*, 2014. <http://www.processing.org/reference/>.
- [5] Wikipedia - Global Positioning System. 2014. <https://en.wikipedia.org/wiki/GPS>.
- [6] Wikipedia - Infrared. 2014. <https://en.wikipedia.org/wiki/Infrared>.
- [7] Wikipedia - Interactive whiteboard. 2014. http://en.wikipedia.org/wiki/Interactive_whiteboard.
- [8] Wikipedia - Ultrasound. 2014. <https://en.wikipedia.org/wiki/Ultrasound>.
- [9] M. Avancini. Using Kinect to emulate an Interactive Whiteboard. 2012. http://latemar.science.unitn.it/segue_userFiles/LITSA/Avancini_Mattia_138793_Thesis.pdf.
- [10] D. Schouten. *Do-it-yourself project to use a Wii remote controller, Wiiscan software and USB powering for a digital interactive whiteboard in classrooms*. <http://wyxs.net/web/wiiscan/>.
- [11] M. Spiegelmock. *Leap Motion Development Essentials*, 2013. ISBN 978-84969-772-9.

Apéndice A

Circuitos

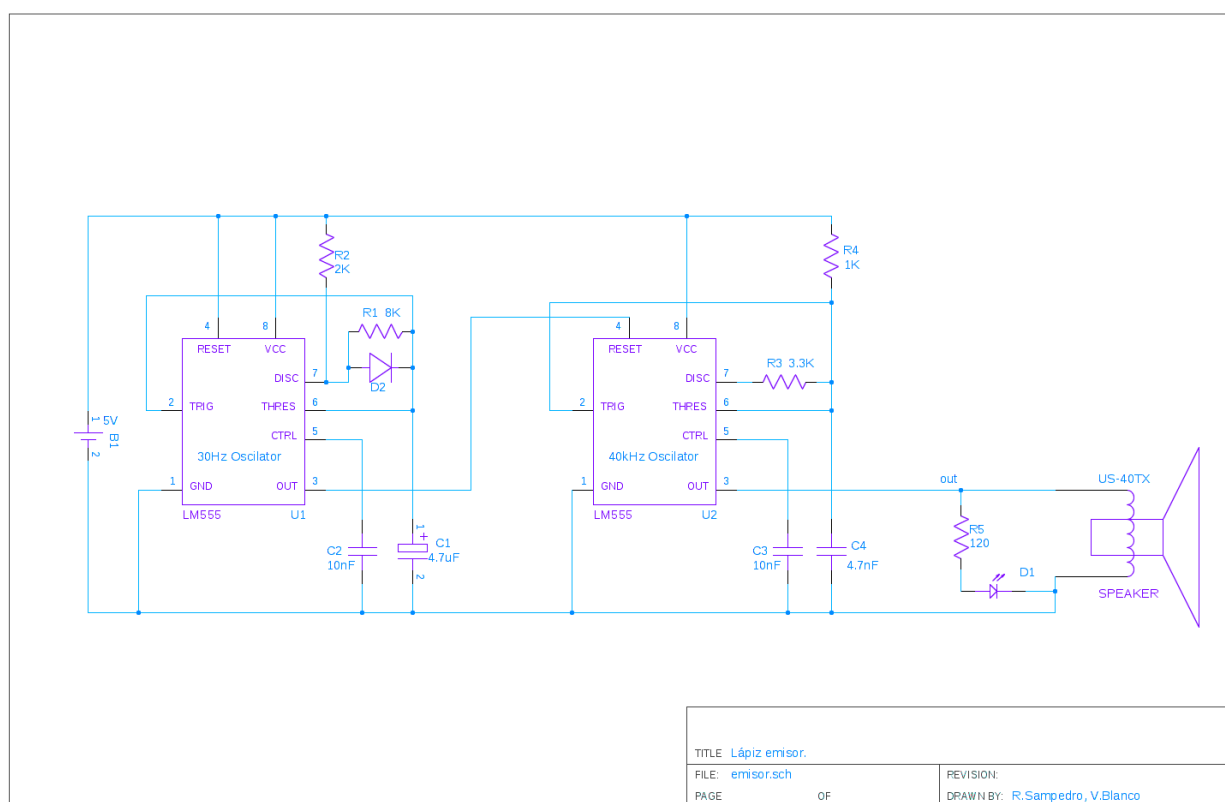


Figura A.1: Circuito de lápiz emisor.

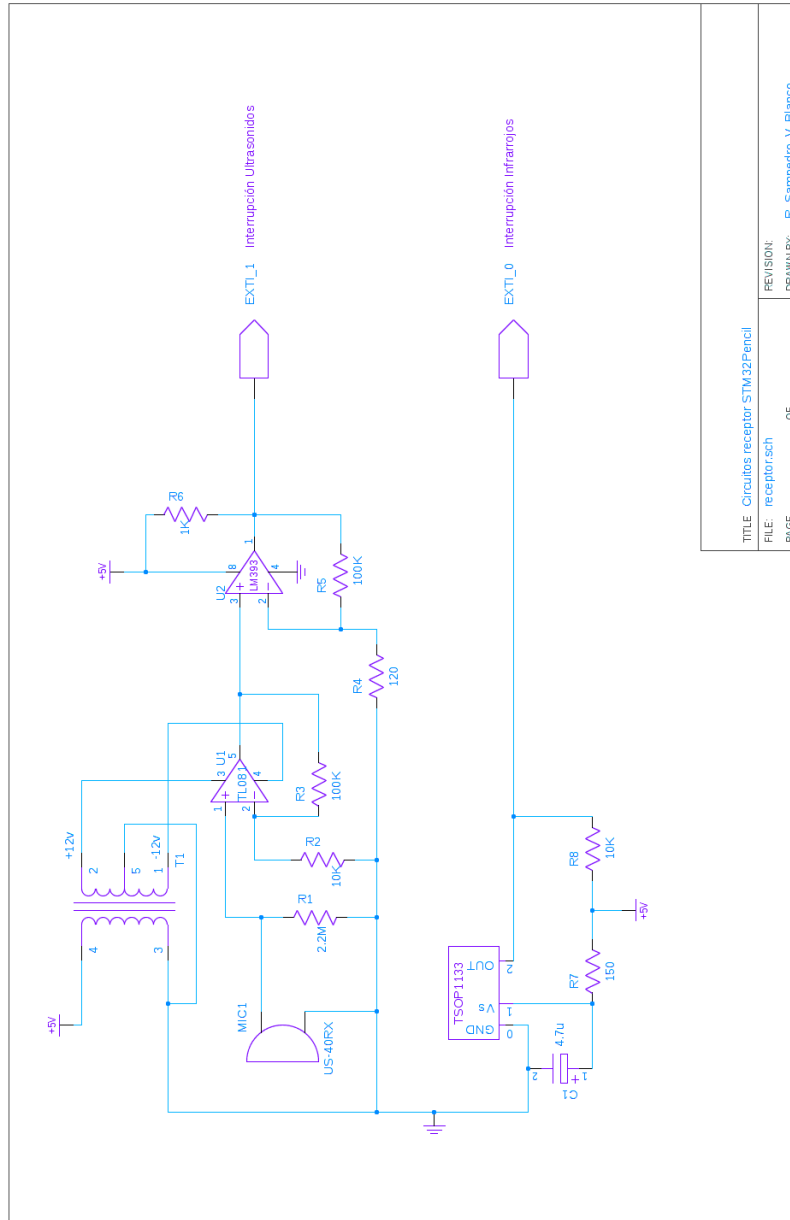


Figura A.2: *Circuito receptor para STM32.*

Apéndice B

Diagramas UML de la aplicación pizarra

A continuación se incluyen una serie de diagramas UML correspondientes a la aplicación pizarra propia desarrollada.

B.1. Diagramas UML de clases

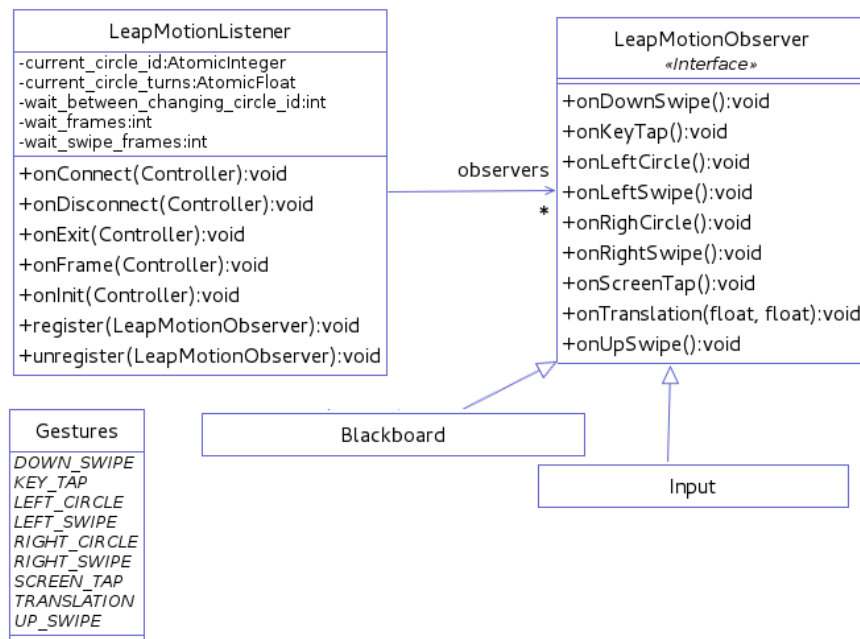


Figura B.1: *Diagrama UML de clases del paquete leapmotion.*

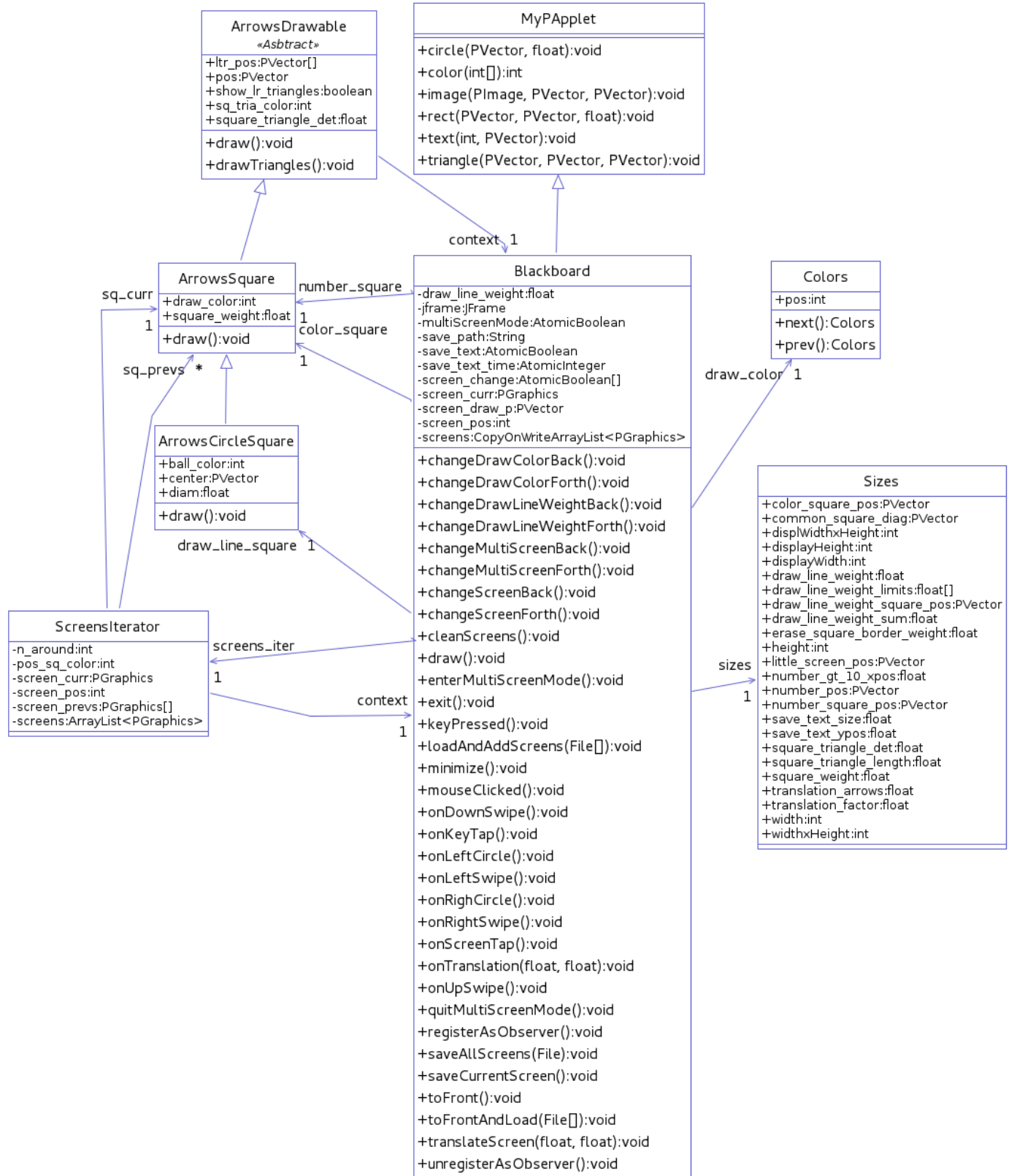


Figura B.2: Diagrama UML de clases del paquete blackboard.

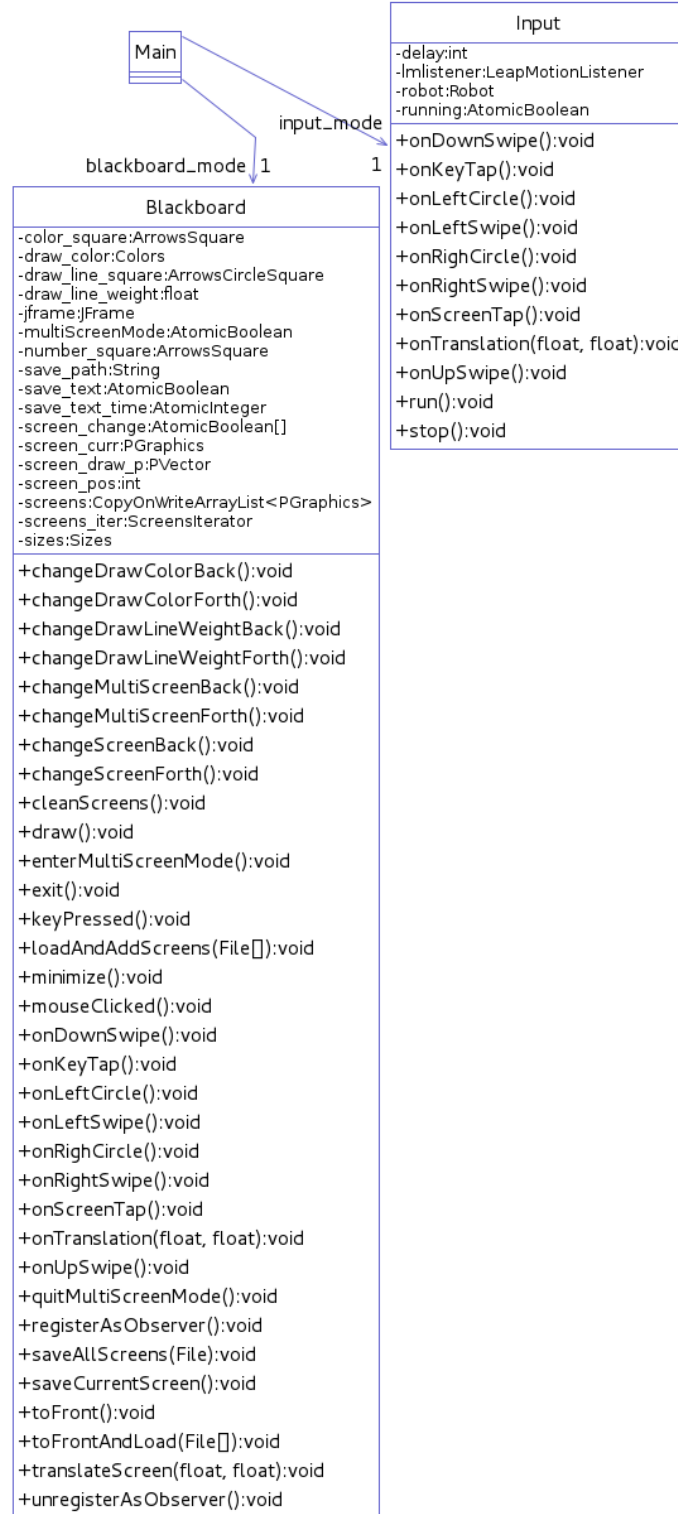


Figura B.3: Diagrama UML de clases del paquete protoboard.

B.2. Diagramas UML de interacción

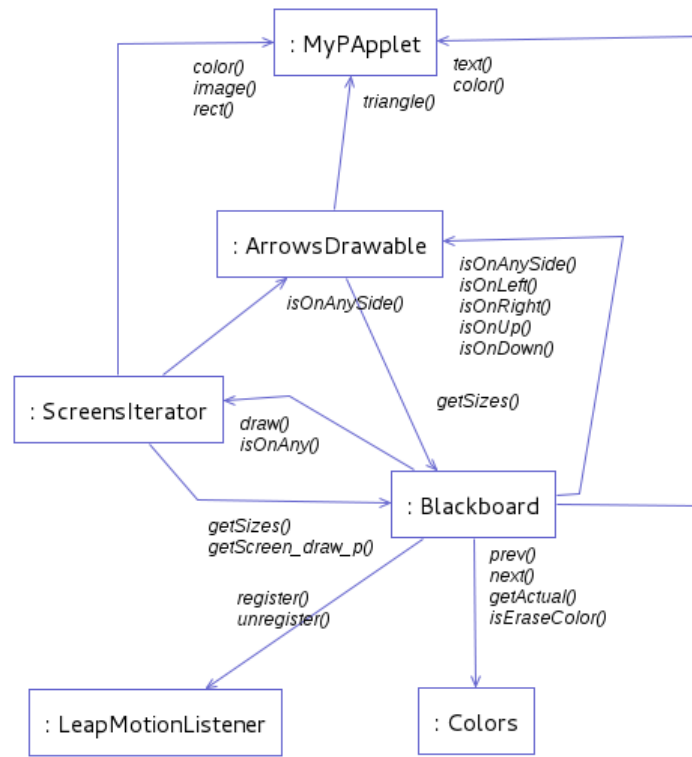


Figura B.4: *Diagrama UML de interacción del paquete blackboard.*

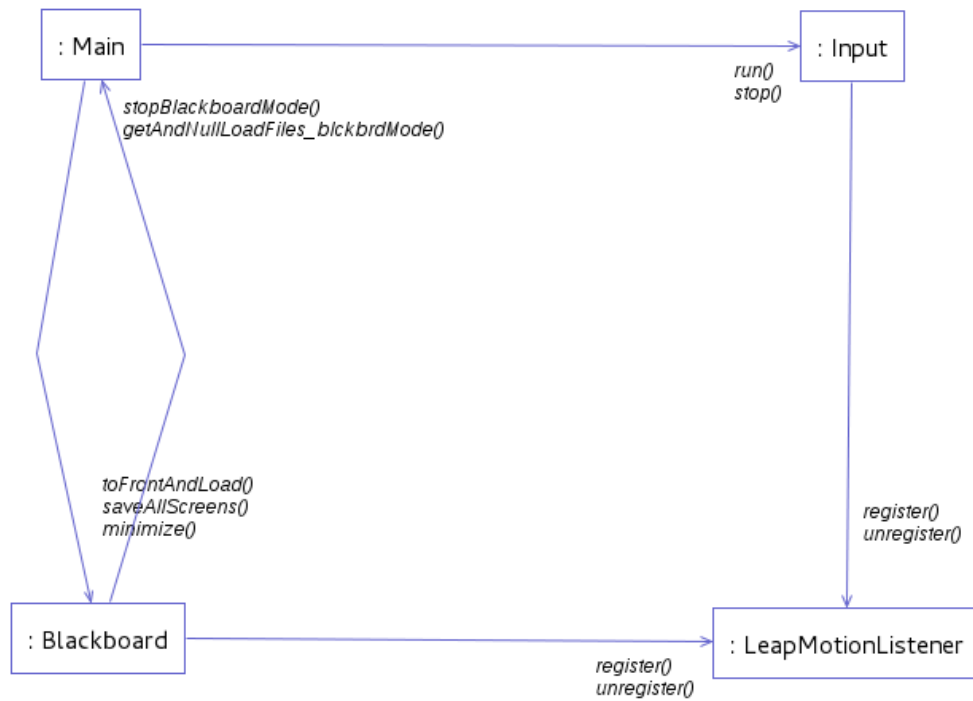


Figura B.5: Diagrama UML de interacción del paquete protoboard.

B.3. Diagrama UML de dependencias entre paquetes

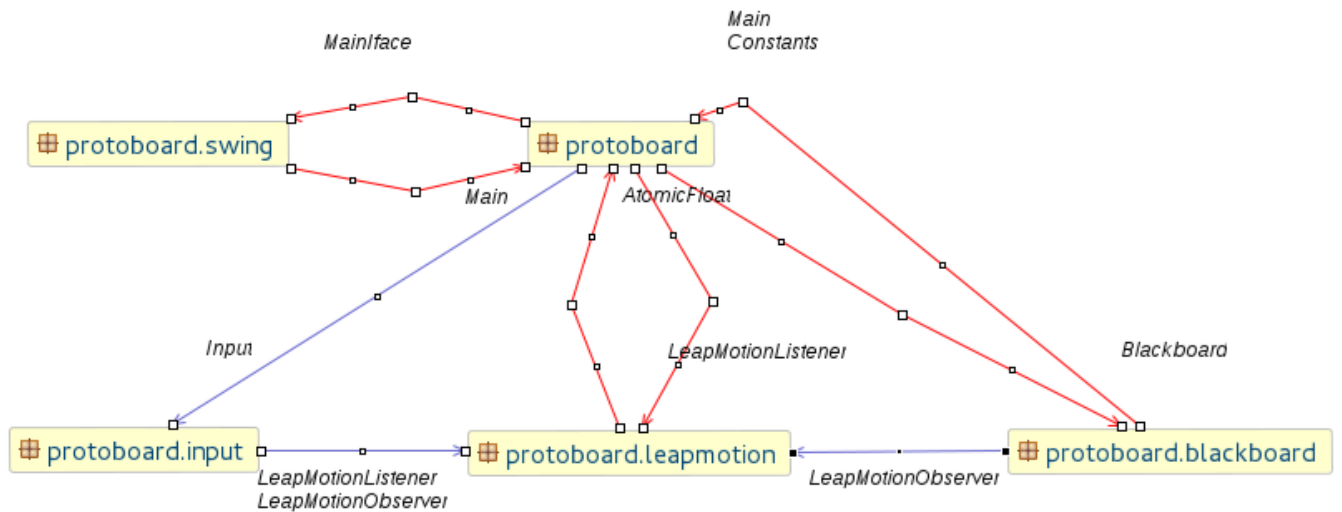


Figura B.6: Diagrama UML de dependencias entre paquetes de la aplicación.

Apéndice C

Reparto de trabajo

C.1. Blanco Gómez, Valentín

A continuación, voy a explicar mi contribución personal al trabajo, en orden cronológico.

Participé en una investigación preliminar sobre el mercado actual de dispositivos sensores 3D, donde encontramos los dispositivos *Leap Motion* y *Kinect* adecuados para una implementación de una pizarra digital. También se investigaron distintas soluciones y aproximaciones al problema del posicionamiento de un puntero (sensores 2D). Fueron explorados distintos softwares existentes que emulan el comportamiento de una pizarra y evaluado su posible uso para nuestro proyecto. Se investigaron y evaluaron las distintas soluciones comerciales existentes en base al precio, funcionalidad, que aparecen en el capítulo 2.

Participé en la evaluación de una solución existente para emular una pizarra digital usando un dispositivo *Wii mote*, a partir del artículo referenciado en [10] y que se profundiza en la sección 3.1, para el cual se montó el dispositivo en un trípode, se montó el resto del equipo necesario (proyector), y se realizaron pruebas de visibilidad de la cámara infrarroja del *Wii mote*, de precisión, manejabilidad... (también descritas en la sección 3.1).

Participé en la evaluación de una solución comercial existente para emular una pizarra digital usando un dispositivo *Kinect*, comercializado por la empresa *Ubi Interactive*, y que se describe en profundidad en la sección 3.2. Se montó el equipo necesario (proyector, *Kinect*) y se procedió a realizar pruebas de fiabilidad, precisión... (descritas en la sección mencionada).

Después de toda la investigación preliminar y evaluación de soluciones existentes, participé en una evaluación global de todos los dispositivos investigados y soluciones software en la que se decidió explorar una solución propia (que se menciona en el capítulo 4). Se decidió explorar el uso de un dispositivo *Leap Motion* para el reconocimiento de gestos, el desarrollo de una solución propia para el posicionamiento del puntero basado en tecnología infrarroja

y ultrasónica, y la prueba en conjunto del software desarrollado para el *Leap Motion* y la solución evaluada que hacía uso del *Wiimote* (que se describe en la sección 4.3).

Después de esta evaluación, se decidió repartir trabajo, y yo me encargué de la parte orientada al software y al uso y programación del dispositivo *Leap Motion* (que se describe en la sección 4.1).

Diseñé la aplicación de pizarra digital, para la cual utilicé el lenguaje de programación *Java* con la librería *Processing* y el *SDK* de *Leap Motion* para *Java*. Consideré el uso de *Processing* por facilitar mucho la programación de entornos que tienen una interfaz gráfica donde el usuario puede pintar, y abstraer toda la programación haciendo uso de *OpenGL* que sería necesaria.

Para la parte orientada a la programación del *Leap Motion*, utilicé como referencias el *Leap Motion API reference*[3] y el libro “*Leap Motion Development Essentials*”, escrito por M. Spiegelmock[11]. Toda esta parte está profundizada en la sección 4.1.

Para la parte orientada a la programación de la propia aplicación pizarra me guíé por la *Processing reference*[4] y por conocimientos propios aprendidos en la asignatura de Informática Gráfica impartida como opcional en este grado. Toda esta parte está descrita en 4.1. Todo el código generado en esta parte está disponible en un repositorio online en *GitHub*¹.

Después del diseño de la aplicación, participé en una evaluación de la experiencia de uso del *Leap Motion*, en la que se probó la aplicación sobre un entorno real (con proyector, *Leap Motion* instalado en la parte de abajo de la pizarra). Se evaluaron los distintos gestos que propuse para la aplicación, su facilidad de uso y aprendizaje. Se evaluó también la experiencia de uso global y la respuesta del programa a gestos mal realizados. A partir de estas pruebas se recogieron datos y mejoras posibles para el programa que después incluí.

¹<http://github.com/valenbg1/protoboard>

Por último, participé en la evaluación de mi aplicación desarrollada junto con la solución evaluada que hacía uso del *Wii mote* (solución descrita en 3.1), la cual se profundiza en la sección 4.3. Probamos posibles interferencias que se pudieran producir entre el *Leap Motion* y el *Wii mote*, y el comportamiento general de mi aplicación en conjunto con un emulador de puntero infrarrojo. A partir de estas pruebas se sacaron conclusiones de uso y mejoras para la aplicación que posteriormente las incluí.

C.2. Sampedro Ruiz, Raúl

En este apéndice voy a explicar cuales son mis principales aportaciones a este proyecto.

En el proceso de consolidación del proyecto participé en la realización de un estudio de mercado, donde pude observar las distintas tecnologías de pizarras digitales y sus fundamentos. En ese estudio, redactado mayoritariamente en el capítulo 2 y 3, pude observar distintas soluciones comerciales de bajo coste basadas en diversas tecnologías como son el caso de: *Ubi Interactive*² usando un *Kinect*, o el trabajo desarrollado por D. Schouten[10] usando un *Wiimote*. A partir de estos datos partimos de la idea de intentar reducir los costes excesivos de otras soluciones comerciales a costes más parecidos a las soluciones de bajo coste sin reducir la precisión.

Queríamos poder emular un ratón con un lápiz digital. Para ello me dediqué a diseñar y teorizar previamente una estrategia de posicionamiento basada en una baliza móvil (que es nuestro lápiz) y una barra de sensores fija para detectar la posición de la baliza. La idea de la solución que apliqué consiste en emitir en la baliza un pulso de ultrasonidos a la vez que se emite un pulso de infrarrojos. Aprovechando la diferencia de velocidades entre la luz y el sonido, comprobé teóricamente que era factible la medida de distancia de la baliza al sensor con una precisión suficiente para la escritura.

Con estos resultados resultaba prometedor continuar con el diseño de la estrategia de posicionamiento, teorizando ahora el método de seguimiento de la baliza. Como ya había comprobado teóricamente que se podía medir distancias con el método antes descrito, mi idea fue replicar periódicamente este comportamiento, consiguiendo así un seguimiento de la posición de la baliza en el tiempo.

²<http://www.ubi-interactive.com/>

En esta nueva aproximación descubrí problemas que se podrían presentar en una posible implementación, como son el caso de los rebotes de ultrasonidos o la máxima distancia que se puede llegar a medir. Estos problemas están comentados y explicados en la sección 4.2.3. Dichos problemas, pese a reducir las posibilidades de la solución inicial, no limitaban lo suficiente la funcionalidad de la solución cómo para tener que replantearla, por lo que decidí seguir adelante con una primera implementación del prototipo.

Para esta implementación inicial necesitaba una pequeña formación en electrónica analógica para poder diseñar los circuitos necesarios. Tras ella diseñé el circuito del lápiz digital (o baliza) que consiste en dos osciladores *LM555* encadenados en modo *aestable* que generan cada 33 ms (30 Hz) un pulso a 40 KHz para el emisor de ultrasonidos y el de infrarrojos. Este circuito está descrito en la figura A.1. Tras comprobar el correcto funcionamiento del emisor con un osciloscopio, empecé a diseñar el circuito de recepción, una primera implementación sin etapa de amplificación, con su posterior comprobación de funcionamiento. Posteriormente se añadió la etapa de amplificación para dar lugar a un prototipo más funcional como el que se puede ver en la figura A.2

Con estos dos circuitos ya terminados pude empezar a programar un micro-controlador (una placa de desarrollo *STM32VDiscovery*) para detectar las señales de la baliza. Esta placa es programada en *bare metal*, es decir sin sistema operativo, para aprovechar al máximo el rendimiento del micro-controlador. Para ello, tras montar un entorno de desarrollo adecuado para la programación en *bare metal*, implementé la descripción del comportamiento del microcontrolador que se encuentra en la sección 4.2.5. El código relativo a esta implementación se puede encontrar aquí³.

³Código microcontrolador - <http://github.com/rsrdesarrollo/STM32Pencil>

Tras ello implementé un pequeño prototipo⁴ en *Processing* para comprobar el correcto funcionamiento de la solución. En esta primera implementación fue donde pude observar los problemas descritos en la sección 5.2 y plantear posibles soluciones.

Con ello y una vez demostrado que es factible en la práctica el posicionamiento en el que se basa mi solución, participé en una prueba de evaluación de usuario con la aplicación de pizarra mencionada en la sección 4.1, conjugando el uso de un *Leap Motion* junto con un *Wiimote*, con el fin de evaluar las interferencias que podrían causarse entre ellos. Los resultados de estas pruebas quedan reflejados en la sección 4.3.

⁴Medida de distancia - <http://gist.github.com/rsrdesarrollo/f41755a86bb92eab6236>

Apéndice D

Introduction

In this chapter we define the concepts of digital whiteboards, also exploring its uses, functionality and disadvantages. In addition we will introduce the new objects positioning and detection devices, and describe an exploration proposal based on this information.

D.1. Digital whiteboard concept

An interactive whiteboard is, broadly speaking, a computer and a surface on which the video is displayed (touch screen, traditional projector...). The distinguishing feature of a digital whiteboard is the existence of a detection mechanism that allows to control a computer (usually the mouse) interacting with the projected image. To accomplish this interaction, typically some kind of pen or even your finger is used. The goal will be replacing the traditional blackboard, significantly expanding its functionality.

The scope of use of a digital whiteboard is very diverse: group workspaces, meetings, conferences... But today, undoubtedly the most common use is in teaching, whatever the level of students is.

Many uses that can be given as^[7]:

- Running software on the computer to be controlled by the digital whiteboard (for



Figura D.1: *Interactive whiteboard at CeBIT 2007.*

example, a web browser).

- Writing and editing notes on the whiteboard that will be saved on your PC.
- Showing written notes with a graphic tablet.
- Controlling the computer from your whiteboard using “*drag and drop*” actions that may produce, for example, notes in different applications or presentations. This opens up a huge range of possible educational purposes, from kindergarten to university.

These *classic* digital whiteboard have a number of disadvantages: they have “limited” functionality because their normal use is reduced to map user actions as movements or mouse clicks on the computer. Another inconvenient is that its sensors have a 2D range, so is not possible to capture some interesting movements like hand gestures, as these sensors are limited to a flat surface. Finally, another major drawback is the price: depending on the technology, there is a wide range of prices, generally expensive. The most affordable systems usually provide only detection mechanisms. For example, we find a system that combines infrared and ultrasonic sensors at a price of 559€¹, but does not include spotlight

¹<http://www.mimioonline.com/p/1/Mimio%20Teach/MimioTeach/>

or projection surface. If we search for interactive whiteboard systems including big touch screen surfaces, the price increases significantly. As an example, a full whiteboard based on infrared technology doubles the previous price (1245\$²); a full touch screen of 65" is even more expensive (4173€³). Also in many of these cases the whiteboard must have a specific installation, which prevents easily carrying it from one place to another; in colleges and universities this implies the need of multiple units.

D.2. New 3D devices

The emergence of new types of sensors that are not limited to a 2D field but allow to detect a depth map using 3D sensors, opens new avenues for development of whiteboards. Significantly expand functionality, allowing to detect gestures, natural movements of the user as “pass a page” or “do zoom to an image”. As examples we can put the *Microsoft Kinect* sensor, which provides depth information, gestures and people recognition, and is intended to be placed at a modest distance so it detects the whole body. Recently, the *Leap Motion* device was commercialized, initially designed for a closer use (beside computer keyboard). It allows 3D hands gestures detection. In section 2.2 and chapter 3 we will do a more detailed discussion of these devices.



Figura D.2: *Leap Motion device.*

²<http://www.goelectronic.com/QOM0+QWB200-PS.html>

³<http://pizarras-digitales.net/pantalla-interactiva-65-4-points>

D.3. Proposed exploration

The project's goal is to design a system that allows converting a projector to a whiteboard. Thus, from the above data, we developed a proposal for an exploration to evaluate different solutions with distinct functionality and analyze their costs. To do this, we focus on three key areas:

- **Devices:** we will analyze the use of different devices and technologies with which to interact with the system (positioning, click and drag actions...).
- **Cost:** we have to evaluate the different solutions always trying to minimize the total cost of the prototype, in order to achieve an inexpensive system.
- **Functionality:** we will value solutions and devices that provide new functionality to the whiteboard idea, such as sensors using 3D depth map.

There will also be necessary to design the software (if possible multi-platform) for the positioning element to be used effectively as a mouse on a computer. We will also consider the development of applications that take advantage of system functionality.

Apéndice E

Conclusions

In this project we explored various alternatives for the development of low cost whiteboards that also add functionality to the existing market. We have seen how, with the output of 3D sensors on the market, considerably extend the capabilities of these devices at affordable prices.

Using one of these devices (*Leap Motion*), we have developed a system that can allow both interaction with the desktop and emulate a blackboard that allows easy storage and sharing all the notes.

In the field of *traditional* whiteboard (in example, 2D positioning the mouse pointer), we have studied alternatives including a very low cost homemade solution from ultrasound and infrared technology. In the remainder of this section we will study the cost of this solution and discuss its possible implementation in real environments.

E.1. About the manufacturing cost

In chapter 2 we saw the current situation of digital whiteboards and its prices. Our main objective since then has been trying to reduce the price as much as possible at a minimum lose of quality.

The material cost in detail in one of our sensors, as seen in figure E.1 is approximately 23€. The breakdown is estimated as follows:

- STM32VDiscovery board - 10€
- 40Hz ultrasonic sensor - 2€
- Infrared sensor - 2€
- Integrated circuits (LM393 y TL081) - 3€
- Analogic devices - 1€
- DC-DC converter - 2€
- 3.3V TTL USB to serial wire - 3€

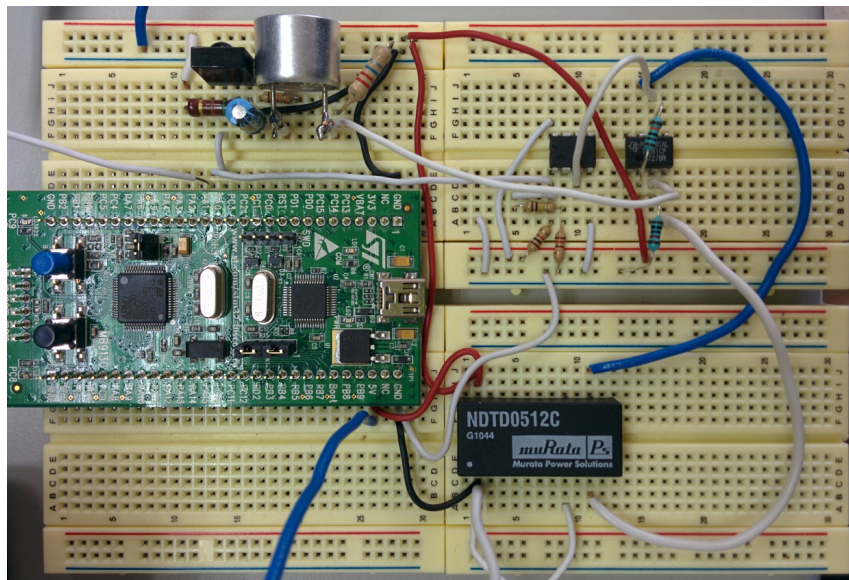


Figura E.1: *Implemented prototyped sensor.*

The most expensive item in this list is the STM32VDiscovery board, which we do not actually completely need. Just the processor is fine, which can be bought separately for less than 5€.

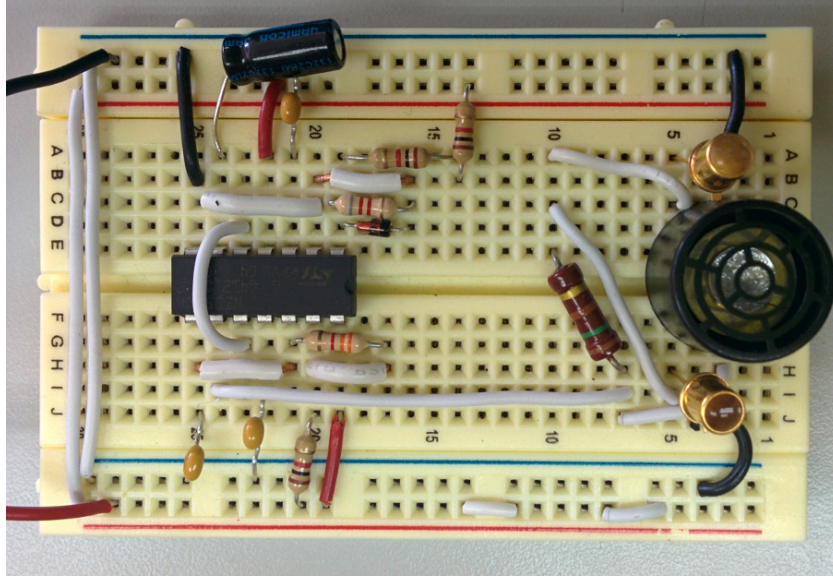


Figura E.2: *Implemented prototyped pencil.*

Our solution to emulate a mouse with these sensors, as seen in section 4.2.1, needs two of these sensors plus a transmitter (as seen in figure 5.2), whose cost does not exceed 4€ retail manufactured.

If we add up all the costs, we estimate a cost of about 50€ in developing the prototype used to position the cursor. While we must note that under a wholesale marketing the cost of manufacturing would be reduced very notably, we can not disregard that the current prototype is to demonstrate the feasibility of the system, and commercial implementation could lead to increased costs in order to improve current faults.

If we add to this the cost of a *Leap Motion* (about 90€), we obtain gesture recognition with a solution just for 140€, compared to prices of the solutions we saw in chapter 2.

We have to keep in mind that these calculations are not assuming a very important cost, which is the cost of a projector and a computer for the use of hardware and software implemented. This clearly exceeds the cost cost of the sensors (many other solutions have this

added cost), but can be neglected, since a projection installation is very common in any classroom. Still, in order to minimize the cost of the computer is very likely to carry the ultrasonic solution without *Leap Motion* to a *Raspberry Pi* system, for example.

E.2. About viability of our positioning solution

Not far from a real solution, our prototype allows us to detect implementation problems about detecting position by ultrasonic and infrared. Below are a list of problems explained and possible solutions:

- **Ultrasonic transmitter being directional:**

Unfortunately we have found ultrasonic transmitters to be highly directional. This means that if the transmitter does not point roughly directly to the sensors, is relatively easy to lose the signal. One possible solution is to use a more omnidirectional transmitter or even use more transmitters pointing in different directions.

- **Loss of precision with noise or attenuation:**

It may occasionally happen that the first oscillation of the ultrasonic receptor is lost or attenuated delaying the ultrasonic wavefront $25\ \mu\text{s}$ (cycle time of ultrasound). It is easy to see that this delay causes a small error in the measurement of approximately 10 mm according to the formulas views in section 4.2.2. For solving this little error, we can either improve the amplification stage in the receiver, or implement a signal treatment in application level.

- **Signal jamming by malicious student:**

It is relatively easy to jam the infrared signal from the transmitter with another transmitter (such as a remote control) correctly directed towards the sensors bar, causing the entire system to be unusable. This problem is hardly treatable because of the very low signal / noise ratio in channel.

E.3. About using digital whiteboards in classrooms

During development of the project we had meetings with teachers for obtaining a list of needs to be met by a digital whiteboard. In this section we report some of them and how they could be covered.

In *Colegio Público Asunción Rincón* teachers have a whiteboard enabled classroom. This has a proximity projector and a resistive surface. The cost of this installation is around 1500€. Teachers use these whiteboards for interactive activities with children, like games of relating figures with names.

For this type of activity very high accuracy is not required, so it could easily be implemented with a 3D sensor as the *Kinect* system.

In *Colegio Gredos San Diego* teachers are interested in being able to interact with the blackboard by writing on it, in order to save what was wrote. This center has shown interest in our approach as it would allow them to position the cursor accurately enough.